Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Žan Anderle

Avtomatsko napovedovanje lastnosti podjetja na podlagi njihove spletne strani

magistrsko delo

magistrski program druge stopnje
računalništvo in informatika

Mentor: izr. prof. dr. Janez Demšar

Ljubljana, 2017

Zahvala

Zahvaljujem se izr. prof. dr. Janezu Demšarju za mentorstvo, strokovno pomoč, podporo, zaupanje in potrpežljivost.
Zahvaljujem se zaposlenim v podjetju Datafy.it za vložen čas in pomoč pri pridobivanju podatkov.
Zahvaljujem se tudi babi Zdravki za skrb in spodbudo. Hvala bratu Kristjanu, ki mi je v vzor in podporo že od malih nog.
Hvala Nataši, ki me je ves čas vztrajno in neomajano podpirala, spodbujala in verjela vame.

In najpomembnejše — hvala staršem. Za vse.

Žan Anderle, 2017

Staršem, Nataši in Sezamčku.

Kazalo

Seznam uporabljenih kratic

kratica	angleško	slovensko
URL	Uniform Resource Locator	Spletni naslov
CRISP-DM	Cross Industry Standard Process for Data Mining	Standardni proces podatkovnega rudarjenja
NAICS	North American Industry Classification System	Severno ameriški klasifikacijski sistem panog
ISIC	International Standard Industrial Classification	Mednarodni standard klasifikacije panog
NLP	Natural language processing	Obdelava naravnega jezika
tf-idf	Term frequency–inverse document frequency	Frekvenca besed - inverzna frekvenca dokumenta
SVM	Support vector machine	Metoda podpornih vektorjev
LDA	Latent Dirichlet Allocation	Latentna Dirichletova alokacija
LSA	Latent semantic Analysis	Latentna semantična analiza
SGD	Stochastic gradient descent	Stohastični gradientni spust
HTML	Hyper text markup language	Jezik za označevanje nadbesedila

Povzetek

Naslov: Avtomatsko napovedovanje lastnosti podjetja na podlagi njihove spletne strani

V magistrskem delu obravnavamo problem napovedovanja lastnosti (panoga, starost, število zaposlenih) podjetja na podlagi njihovega spletnega mesta. Predlagamo več napovednih modelov, ki spletno mesto obravnavajo na različne načine. V delu pokažemo kako iz spletnega mesta izluščiti tiste značilke, ki bodo za neko specifično napoved uporabne. V našem primeru se za najbolj uporabno izkaže besedilo celotnega spletnega mesta ter besedilo, ki ga najdemo v meta oznakah. S tem dobimo dva ločena napovedna modela, ki ju lahko združimo v eno združeno napoved. Tak združevalni napovedni model smo uporabili pri napovedovanju panoge podjetja, kjer je dosegel zadovoljive rezultate. Obenem smo preizkusili tudi napovedovanje na podlagi meta značilk spletnega mesta, s katerimi lahko spletno mesto opišemo na drugačen način in se s tem izognemo računsko zahtevni obdelavi besedil. Ta model smo preizkusili na problemu napovedovanja starosti in števila zaposlenih v podjetju. Z modelom nismo dosegli zadovoljivih rezultatov. V delu raziščemo tudi problematiko primernega nabora podatkov za razvijanje napovednih modelov, ki se za napoved zanašajo na spletna mesta. Ugotovimo, da je ta problematičen korak ključen za doseganje boljših rezultatov.

Ključne besede

klasifikacija spletnih mest, strojno učenje, informacije spletnih strani

Abstract

Title: Automatic prediction of company's characteristics based on their website

Our main objective is predicting company's characteristics (industry, age, number of employees) based on the company's website. We present different prediction models which all extract information from the website in distinct ways. We show what features to extract from a website, that will be useful for a specific prediction. We find that website's content text and meta tags text are often the most relevant. By using these texts we get two separate prediction models and we can also use them in an ensemble model. The latter was used in predicting the company's industry and achieved satisfactory results. We also tested using alternative ways to describe a website by using different meta data that we can extract from a website. This is useful when it is necessary to avoid the computational cost of performing text analysis. We used a model using these features in predicting the age and number of employees. The model was not particularly successful. We also discuss the problem of an appropriate dataset needed for developing aformentioned prediction models. We find that solving this problem is crucial for achieving better results.

Keywords

website classification, machine learning, website information

Poglavje 1

Uvod

V poslovnem svetu, kjer se konstantno išče nove stranke in nove možnosti sodelovanja, so podatki o podjetjih zelo dragoceni. Lastnosti podjetja, kot so panoga, število zaposlenih, prihodki in starost podjetja, so le ena vrsta podatkov, ki se izkažejo za zelo uporabne. Podjetja na podlagi teh lastnosti lahko iščejo potencialne stranke, nove poslovne partnerje, ali le raziskujejo trg za nove poslovne možnosti. Zaradi tega je zanašanje na le en vir teh podatkov lahko nezanesljivo. Profesionalna socialna omrežja, kot so LinkedIn[1], Xing[2], Viadeo[3], razpolagajo s podatki o posameznikih in podjetjih, vendar so ti podatki velikokrat okrnjeni, niso prosto dostopni ali celo napačni. Zato napovedovanje teh lastnosti na podlagi prosto dostopnih virov predstavlja veliko konkurenčno prednost.

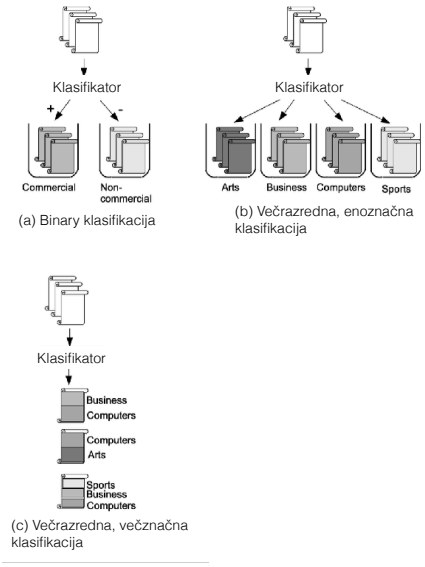
Splet je postal prvo mesto, kjer si stranke poiščejo informacije o podjetju. Posledično so za podjetja spletna mesta že nekaj časa ključnega pomena. To pomeni, da imajo v veliki meri podjetja svoja spletišča, kjer podajo vsaj neke osnovne informacije o svojem delovanju. Nekatere spletne strani vsebujejo tudi prav tiste podatke, ki nas v tem magistrskem delu zanimajo. Vendar za nas to ni bistveno, saj ni zanesljivo, da bodo dotični podatki zares prisotni. Namesto tega se osredotočamo na spletišče kot tako. Zanimajo nas

[1] https://www.linkedin.com/
[2] https://www.xing.com/
[3] http://us.viadeo.com/en/

podatki, ki jih lahko razberemo s spletne strani; podatki kot so velikost spletnega mesta, vsebina celotnega spletnega mesta, število povezav na spletnem mestu, itd. Na podlagi teh prosto dostopnih podatkov smo želeli čim boljše napovedati nekatere lastnosti podjetja. Osnovna predpostavka je, da je želja podjetja na lastnem spletnem mestu predstaviti čim bolj relevantne podatke in da lahko zaradi tega mi s spletišča razberemo relevantne lastnosti.

1.1 Pregled literature

Literature na temo napovedovanja določenih lastnosti podjetja na podlagi njihovega spletnega mesta skoraj ne najdemo. Izjema je raziskava [1], v kateri John M. Pierre poskuša na podlagi spletnega mesta napovedati panogo podjetja, kar je tudi eden izmed pomembnejših ciljev tega magistrskega dela. K tej raziskavi se bomo še vrnili in si jo natančneje ogledali. Razen omenjene raziskave, podobnih del ne najdemo. Tudi podobnih problemov, ki bi recimo na podlagi spletnega mesta obravnavali napovedovanje neke lastnosti lastnika tega mesta, ni veliko. Najdemo recimo samo raziskavo, ki sta jo opravila Dirk Thorleuchter in Dirk Van den Poel [2], v kateri napovedujeta uspeh prodajnega spletnega mesta na podlagi vsebine tega spletnega mesta. Kljub temu pa najdemo precej literature na temo klasifikacije spletnih strani in spletnih mest [3, 4, 5, 6, 7, 8, 9, 10, 11, 1, 12, 13]. Načinov klasifikacije spletnih strani je ogromno. V grobem jih lahko razdelimo na število razredov v katere strani razvrščamo, torej na binarno in večrazredno klasifikacijo, kot lahko vidimo na Sliki 1.1. Primer uporabe binarne klasifikacije bi bil recimo ugotavljanje ali je določena spletna stran zlonamerna ali ne [13]. Večrazredno klasifikacijo še dodatno razvrščamo glede na število razredov, ki jih lahko pripišemo enemu vzorcu; eno- ali večznačna klasifikacija. Pregled obstoječih raziskav vseh različnih klasifikacij, ki smo jih našteli, so opravili Qi, Xiaoguang in Davison, Brian D [7]. Pri tem pregledajo najbolj pogosto uporabljene tehnike za posamezne klasifikacije ter primerjajo njihovo učinkovitost. Glede na to, da se naše delo še najbolj približa problemu enoznačne večrazredne



Slika 1.1: Razvrščanje klasifikacije spletnih strani. Povzeto po [7]

.

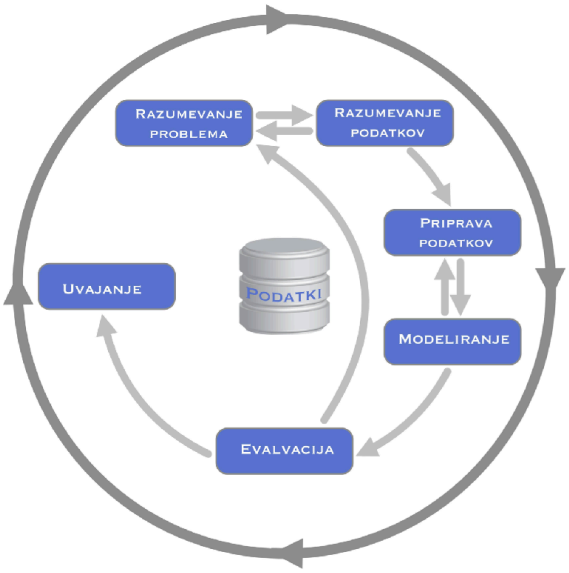
klasifikacije, smo si natančneje pogledali raziskave na tem področju. Omenjena raziskava [7] opaža tudi, da se klasifikacija vsebin na spletu močno opira na raziskave narejene na klasifikacijo besedil. Zaradi tega smo si natančneje pogledali tudi nekaj sorodnih raziskav v domeni besedil. Hemant Misra, François Yvon, Olivier Cappé in Joemon Jose [14] raziskujejo segmentacijo besedil in dosežejo dobre rezultate z uporabo LDA. Ivan Vulić, Wim De Smet, Jie Tang in Marie-Francine Moens [15] prav tako z uporabo LDA dosežejo dobre rezultate pri klasifikaciji večjezičnih besedil. To je za nas zanimivo, saj bomo tudi v tem delu rokovali z večjezičnimi podatki. Uporabnost dodajanja LDA vektorja k obstoječim značilkam vreče besed raziskujejo Jorge Victor Carrera, Trejo, Grigori, Sidorov, Sabino, Miranda-Jiménez, Marco Moreno, Ibarra in Rodrigo Cadena, Martínez [16]. Omenjena dela na področju besedilne analize nam nakazujejo, da bi bilo v našem modelu smiselno preizkusiti tudi LDA za izboljšavo rezultatov.

Klasifikacijo spletnih vsebin poleg že omenjenih razvrstitev delimo tudi na klasifikacijo spletnih strani in klasifikacijo spletnih mest. Naše delo se

uvršča v slednjo kategorijo. Seveda nam kljub temu raziskave na področju klasifikacije spletnih strani [4, 5, 7, 8, 9, 13] lahko veliko pripomorejo pri iskanju ustreznih rešitev. Qi, Xiaoguang in Davison, Brian D [7] so se v že omenjenem delu osredotočali predvsem na klasifikacijo spletnih strani, kar nam je dalo navdih za naš osnovni napovedni model, ki spletno mesto obravnava besedilno. Shen, Dou, Chen, Zheng, Yang, Qiang, Zeng, Hua-Jun, Zhang, Benyu, Lu, Yuchang, Ma, Wei-Ying [8] raziskujejo uporabo drugih HTML struktur za pridobivanje besedila, ki ga uporabimo kot osnovo za klasifikacijo. Prav tako pokažejo uporabnost LSA. Najboljše rezultate so dobili z uporabo naslovov, meta opisov in meta ključnih besed. Rung-Ching Chen in Chung-Hsun Hsieh [9] pregledata uporabnost algoritma, ki preko glasovanja združi rezultate več klasifikatorjev. Tudi v njunem delu je prikazana uporabnost LSA pri klasifikaciji spletnih strani.
Raziskave, ki se osredotočajo specifično na klasifikacijo spletnih mest [3, 6, 10, 11, 1, 12], nam dajo vpogled v dodatne možnosti napovednih modelov in možnosti za nadaljne raziskave.

Več raziskav za klasifikacijo tako spletnih strani kot spletnih mest raziskuje uporabo zgolj naslovov URL [4, 5, 6] in pri tem dosegajo rezultate, ki so primerljivi s tistimi, ki analizirajo celotne strani/mesta, a se pri tem izognejo potrebi po pridobivanju vsebine spletnih strani/mest.
Jean-Charles Lamirel in David Reymond [12] v svojem delu dosežeta dobre rezultate klasifikacije spletnih mest zgolj na podlagi vsebine navigacijskih menijev. Podobno kot napovedovanje na podlagi meta oznak je tak model uporaben, ker nam omogoča, da se izognemo obdelavi celotnega spletnega mesta, kar je pri veliki količini podatkov računsko precej zahtevna operacija. Bauer, Christian in Scharl, Arno [17] raziščeta kako lahko spletna mesta opišemo z različnimi značilkami, ki jih iz spletnih mest uspemo precej preprosto izluščiti. Take značilke nam lahko predstavljajo alternativno podlago za naše napovedne modele.

Oglejmo si še raziskavo, ki je našemu delu še najbolj sorodna. John M. Pierre [1] je v raziskavi na podlagi spletnega mesta podjetja želel napovedati



Slika 1.2: Diagram metodologije CRISP-DM. Povzeto po [19]

.

panogo, ki ji podjetje pripada, kar je tudi eden izmed ciljev tega magistrskega dela. Za nabor spletnih mest podjetij so morali poskrbeti sami. Uporabljene tehnike pridobivanja spletnih mest so bile podobne kot v tem magistrskem delu. Za razvrstitev panog so uporabili sistem NAICS [18] in nabor podatkov je vključeval le ameriška podjetja. Napovedni model je deloval na podlagi besedil najdenih v spletnem mestu. Primerjal je dva različna načina pridobivanja besedil — celotno besedil in besedilo najdeno v meta oznakah. Slednji način se je izkazal za bolj uspešnega. Njihov napovedni model je uspel doseči povprečen priklic 75%.

V magistrskem delu smo želeli preizkusiti podobne modele kot v tej raziskavi in obenem razviti model, ki bi dobre rezultate dosegal ne glede na državo podjetja.

1.2 Metodologija

Problema napovedovanja lastnosti podjetja smo se lotili po metodologiji CRISP-DM [19], ki je podrobneje prikazan na sliki 1.2. Najprej smo morali poskrbeti za ustrezen nabor podatkov, ki so osnova za razvijanje in testiranje napovednih modelov. Ker je naše napovedovanje zelo specifično, obstoječih naborov podatkov, ki bi bili primerni, nismo našli. Zato smo nabor podatkov ustvarili s pridobivanjem podatkov iz prosto dostopnih virov na spletu. Podatke smo najprej očistili; odstranili vse vzorce, ki zaradi raznoraznih nepravilnosti niso bili primerni. Bolj podrobno je obdelava podatkov predstavljena v Poglavju 2. Za tem smo jih primerno obdelali in pripravili za uporabo v algoritmih strojnega učenja. Da bi poiskali napovedni model, ki bi željene lastnosti podjetja napovedal čim bolj natančno, smo preizkusili več različnih možnost z različnimi parametri. Vse opisano smo razvili v programskem jeziku Python s pomočjo orodij *iPython Notebook* [20] in *scikit-learn* [21].

1.3 Struktura

Magistrsko delo je sestavljeno iz 5 poglavij. Najprej so predstavljeni podatki s katerimi smo delovali, kako smo jih pridobili in kako smo jih kasneje očistili in obdelali. V Poglavju 3 so predstavljeni različni modeli, ki smo jih uporabili za napovedovanje lastnosti. Rezultati in primerjava le-teh so predstavljeni v Poglavju 4, v Poglavju 5 pa so predstavljene sklepne ugotovitve in možnosti za nadaljne delo.

Poglavje 2

Podatki

Izbira in določanje učinkovitega napovednega modela temelji na testnih podatkih. Zato je pravilna izbira in obdelava podatkov, ki jih bomo uporabili za določanje modela, ključnega pomena. Za naše testne podatke potrebujemo seznam podjetij, njihova spletna mesta in lastnosti, ki jih želimo napovedati. Torej, da bi napovedali lastnosti omenjene v uvodu, za vsako podjetje v našem naboru podatkov poleg spletnega mesta potrebujemo informacijo o

- panogi, ki ji podjetje pripada,
- starosti podjetja,
- prihodkih podjetja in
- številu zaposlenih.

2.1 Vir podatkov

Ker obstoječega nabora podatkov, ki bi ustrezal našim zahtevam, nismo našli, smo morali za primeren nabor podatkov poskrbeti sami. Podatki, ki smo jih uporabili v tem magistrskem delu, temeljijo na prosto dostopnih podatkih pridobljenih s spleta. Pridobljeni so bili na profesionalnih socialnih omrežjih, kot so Linkedin, Xing in Viadeo. Na teh omrežjih podjetja sama podajo informacijo o panogi, starosti, številu zaposlenih in povezavo do spletnega mesta.

Podatek o prihodkih podjetja ni prosto dostopen ne na teh omrežjih kot tudi nikjer drugje. Za posamezne države in za zadosti velika podjetja se lahko pridobi tudi ta podatek iz drugih virov, vendar ker naš nabor namenoma vsebuje podjetja različnih velikosti in iz različnih držav, je podatek o prihodkih za nas ostal nedostopen. Posledica pomanjkanja teh podatkov je, da napovedi o prihodkih podjetja nismo mogli izvesti.

Ker cilj magistrskega dela ni specifičen samo za eno državo ali za samo eno vrsto podjetij, smo želeli, da je nabor podatkov glede na državo in vse lastnosti podjetja čim bolj raznovrsten. Obenem pa je za uspešno delovanje napovednih modelov ključnega pomena tudi sama količina podatkov. Podatki so bili pridobljeni v sodelovanju s podjetjem Datafy.it [1] in so zadostili tako pogoju raznovrstnosti kot tudi zadostni količini. Obenem je tako sodelovanje omogočalo tudi preverjanje našega modela na konkretnem, realnem problemu v poslovnem svetu.

2.2 Spletno mesto

Spletno mesto podjetja kot podatek je bil med najpomembnejšimi ter med najbolj zahtevnimi od vseh podatkov, s katerimi smo rokovali. Med najpomembnejšimi saj naša celotna napoved temelji na njem. Med najbolj zahtevnimi pa ker spletna mesta niso enolična in ker so obsežna. Posledično je težko določiti kaj smatramo kot veljavno spletno mesto in kaj ne. Od podjetja do podjetja se močno razlikujejo, zaradi česar je tudi težko zaznati, kaj bi bile lahko pogoste neveljavnosti (za naš nabor podatkov).
Poglejmo, kako lahko smatramo spletno mesto ali posamezno spletno stran podjetja kot neveljavno za naš nabor podatkov. Podjetja na profesionalnih socialnih omrežjih podajo povezavo do svojega spletnega mesta. Predpostavka, ki smo jo tu morali narediti, je, da je ta povezava delujoča in da zares kaže na spletno mesto podjetja. V fazi čiščenja podatkov se je izkazalo, da ta predpostavka ne drži vedno. Take vzorce smo smatrali za neveljavne

[1] https://datafy.it/

in jih odstranili iz nabora. Pogosti primeri takih problemov so bili:

- nedelujoča povezava,
- povezava, ki kaže na stran, ki je potekla veljavnost domene in posledično pod povezavo najdemo obvestilo ponudnika registracije domene ter
- povezava, ki kaže na eno izmed socialnih omrežij (LinkedIn, Facebook, Twitter, itd.).

Začetnih vzorcev je bilo 289, 779. Pri omenjenem čiščenju smo odstranili 21, 612 vzorcev z nedelujočo povezavo in pretečeno domeno ter 7, 123 vzorcev s povezavo na eno izmed socialnih omrežij.

V primeru veljavne povezave smo celotno spletno mesto shranili na lokalen trdi disk, s čimer smo zagotovili ponovljivost načrtovanja in testiranja modelov. Spletno mesto smo pridobili s pomočjo rekurzivnega sledenja vsem povezavam najdenih na prvotni strani, ki so kazale na isto domeno. To je najbolj osnoven način pridobitve spletnega mesta iz povezave, a s tem načinom ni nikoli zagotovila, da smo zares dobili vse spletne strani tega mesta. Vendar za naše potrebe to ni predstavljalo problema, saj smo za uspešen rezultat morali tudi zagotoviti, da bomo lahko postopek ponovili za vsak nadaljni vzorec, ki ni del našega nabora podatkov.

Kot so ugotavljali tudi v sorodnih raziskavah [1], nam ne glede na čiščenje ostane problem raznolikosti in velike stopnje šuma v različnih spletnih mestih. To je problematično predvsem zato, ker se večina raziskav pri klasifikaciji zanaša na kvaliteten, urejen nabor podatkov za učenje modelov. Ker takega kvalitetnega nabora podatkov pri spletnih mestih ne gre za pričakovati, morajo biti naši napovedni modeli precej robustni.

2.3 Panoga

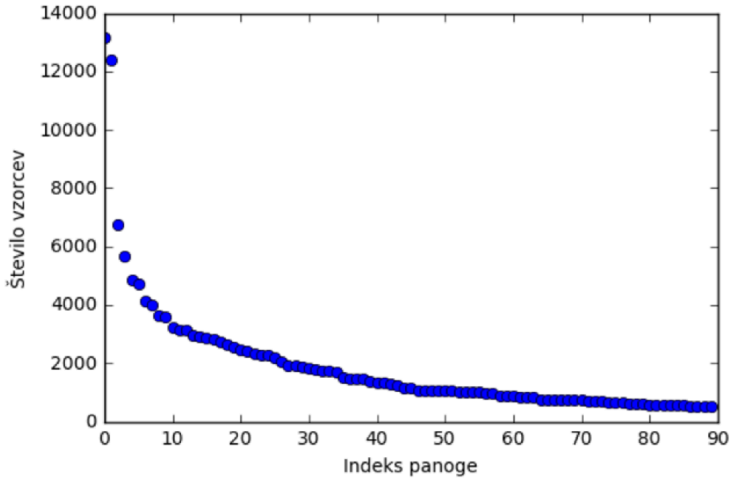
Podatki o panogi podjetja so mnogo manj problematični kot spletna mesta, vendar velikokrat bolj subjektivni. Podjetja si na profesionalnih socialnih

omrežjih sama določajo panogo kateri pripadajo in izberejo lahko le eno. Posledica tega je, da je za specifično podjetje težko objektivno vedeti kateri panogi najbolj ustreza. Tudi primeri, kjer bi bilo teh panog lahko več, niso pokriti. Primorani smo dano panogo vzeti za edino in najbolj primerno, tudi če bi bila katera druga bolj primerna. Po drugi strani pa nam ta način do neke mere poenostavi problem, saj predpostavljamo, da lahko eno podjetje pripada le eni panogi. To pomeni, da je naš večrazredni klasifikacijski problem enoznačen in ne večznačen.

Seznam vseh možnih panog, ki so uporabljene v tem magistrskem delu, je seznam, ki ga določa Linkedin [22]. Glavni razlog, da smo se odločili za to razvrstitev panog in ne katero drugo, recimo NAICS [18] ali ISIC [23], je da so naši podatki že sledili tej razporeditvi. Če bi želeli prevesti te panoge na katero drugo razvrstitev, bi bili rezultati le približki tega, kar je podjetje navedlo kot panogo. S tem bi se lahko še bolj oddaljili od "dejanske"panoge. Če pa bi prevedli na katero koli od manj obsežnih razporeditev, bi s tem izgubili nivo resolucije in točnosti, ki nam jih izbrana razporeditev daje. V vsakem primeru bi s kakršno koli izboljšavo rezultatov, kot posledico izbora alternativne razporeditve panog, izgubili točnost izbrane panoge za posamezno podjetje.

Ker smo za podatke uporabili resnična podjetja, se to odraža tudi v neenakomerni razporeditvi vzorcev glede na panogo. Nekatere panoge so že v osnovi bolj zastopane, dodatno pa se odraža tudi dejstvo, da opazujemo le podjetja, ki so zadosti v koraku s časom, da imajo lastno spletno mesto in so predstavljena na profesionalnih socialnih omrežjih. Posledično so nekatere panoge veliko bolj zastopane, medtem ko imajo nekatere panoge nesorazmerno malo vzorcev. Vse vzorce, ki pripadajo premalo zastopanim panogam (500 vzorcev ali manj), smo iz našega nabora podatkov odstranili. Mejo smo postavili empirično glede na nabor podatkov.

Končno razporeditev vzorcev glede na panogo lahko vidimo na Sliki 2.1, kjer je lepo razvidna tudi neuravnoteženost našega nabora podatkov.



Slika 2.1: Razporeditev vzorcev glede na panogo

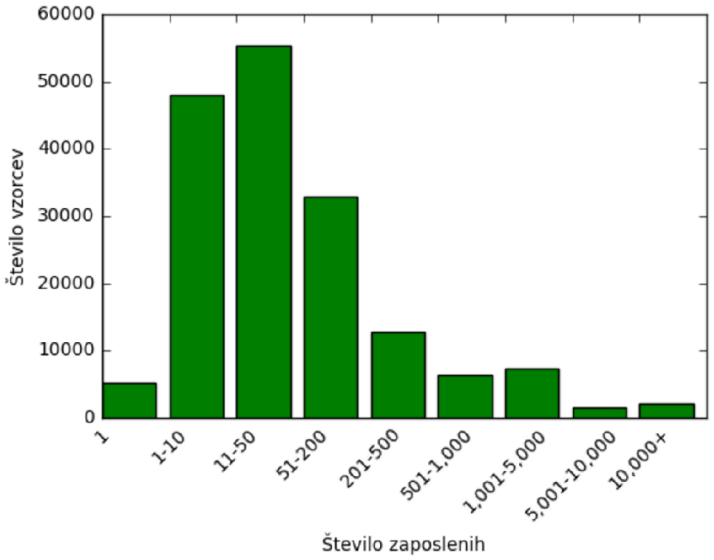
2.4 Starost

Podatek o starosti podjetja smo predstavili kot podatek o letu ustanovitve podjetja. Pri tem je bilo potrebno le rutinsko čiščenje podatkov – na primer odstranitev vzorcev, ki so imeli leto ustanovitve v prihodnosti.

2.5 Število zaposlenih

Podatek o številu zaposlenih smo lahko izluščili le kot interval, kateremu pripada posamezno podjetje (npr. 11-50 zaposlenih). Vsakemu podjetju smo tako lahko pripisali enega izmed naslednjih intervalov:

- 1 zaposlen,
- 1-10 zaposlenih,
- 11-50 zaposlenih,
- 51-200 zaposlenih,



Slika 2.2: Razporeditev vzorcev glede na število zaposlenih

- 201-500 zaposlenih,
- 501-1000 zaposlenih,
- 1001-5000 zaposlenih,
- 5001-10000 zaposlenih in
- več kot 10000 zaposlenih.

Razporeditev vzorcev, ki je tudi v tem primeru neuravnotežena, lahko vidimo na histogramu na Sliki 2.2.

Torej spremenljivko, ki bi jo ponavadi obravnavali kot zvezno, smo dobili na diskreten način. To ni problematično, saj nas tudi kot rezultat ne zanima točno število zaposlenih, vendar le približna velikost podjetja (veliko, malo, srednje). Vseeno pa smo morali pri pripravi podatkov poenotiti intervale. Ker tudi ta podatek podjetja ročno vnesejo in ker se intervali razlikujejo od vira do vira, smo morali podatke smiselno združiti v skupne intervale iz različnih intervalov in iz različnih načinov podajanja števila zaposlenih.

2.6 Uporaba podatkov

Za uspešno in pravilno izdelavo napovednih modelov smo podatke razdelili na dve množici:

- učna množica,
- testna množica.

Podatke smo razdelili naključno in v razmerju 80/20. Ker smo poskrbeli za zadostno količino podatkov, določanje tega razmerja ni bilo bistveno. Učno množico smo uporabili za določanje smiselnih značilk in za tem, s pomočjo prečnega preverjanja, določanje optimalnih parametrov. Testno množico smo uporabili za preverjanje učinkovitosti naših modelov.

Poglavje 3

Napovedovanje

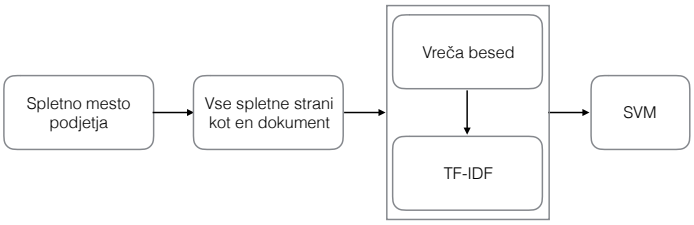
V nadaljevanju bomo predstavili napovedne modele za napovedovanje treh lastnosti podjetja: panoga, starost in število zaposlenih. Kot že omenjeno v Poglavju 2 smo napovedovanje prihodkov podjetja izpustili, saj nismo uspeli zbrati primernih prosto dostopnih podatkov. Vsak model je bil načrtovan ločeno za vsako od treh napovedi. Modeli so si deloma tudi podobni. Skupna točka vseh pa je, da za napoved potrebujejo le spletno mesto podjetja, iz katere potem izluščijo potrebne značilke.

Vsi napovedni modeli so bili implementirani v programskem jeziku Python v okolju *iPython Notebook* [20]. Za ekstrakcijo značilk iz spletnih mest in posameznih spletnih strani smo si pomagali s knjižnjico *Beautiful Soup 4*[1]. Za razvoj ostalih delov napovednega modela smo uporabili programski paket *scikit-learn* [21], ki vsebuje implementacije vseh algoritmov NLP-ja in strojnega učenja, ki smo jih pri tem delu potrebovali.

Kot smo si ogledali v Poglavju 1.1, raziskav, ki bi obravnavale napovedovanje lastnosti podjetja na podlagi spletnega mesta, skoraj nismo našli. Vseeno pa smo si pri načrtovanju modelov lahko pomagali z vsemi raziskavami, ki obravnavajo sorodne probleme klasifikacije spletnih mest in spletnih strani.

Od vseh napovednih modelov smo se najbolj osredotočili na napovedo-

[1] https://www.crummy.com/software/BeautifulSoup/bs4/doc/

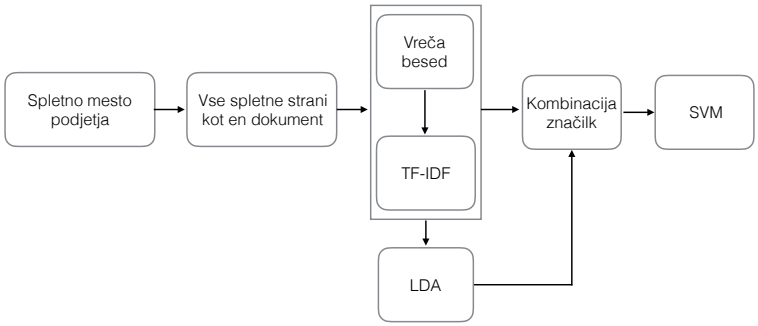


Slika 3.1: Diagram delovanja napovednega modela

vanje panoge, saj je tudi v poslovnem svetu ta podatek velikokrat najbolj zanimiv; preden se sprašujemo, koliko je podjetje staro in veliko, nas zanima s čim se sploh ukvarja.

3.1 Panoga

Pri prvotnem izboru modela smo se oprli na že obstoječe raziskave. Problem napovedovanja panoge glede na spletno mesto sicer najdemo le v eni raziskavi [1], a nam ta poda odlična izhodišča. Najdemo tudi sorodne probleme, ki so nam pri načrtovanju modela predstavljali dodatne oporne točke. Na podlagi pregleda različnih raziskav klasifikacije spletnih strani [7] ugotovimo, da je za osnovni model smiselno izbrati model, ki problem obravnava podobno klasifikaciji besedil. Naš osnovni model vse spletne strani posameznega spletišča prebere kot en dokument, nad katerim potem izvede klasične tehnike NLP-ja. V osnovnem primeru je bila to vreča besed in tf-idf. Vektorje, ki jih na ta način dobimo, potem obravnavamo kot značilke. Za napovedni algoritem smo izbrali SVM. Diagram celotnega modela lahko vidimo



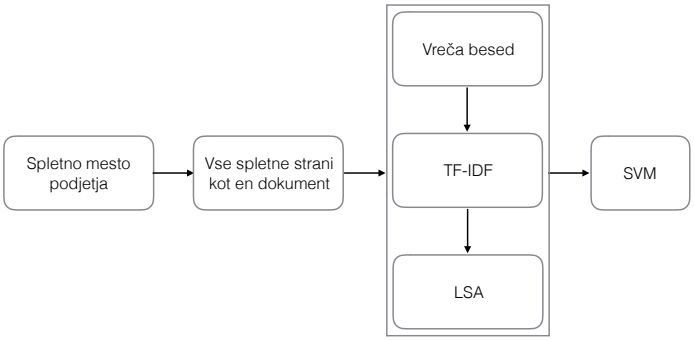
Slika 3.2: Diagram delovanja napovednega modela s kombinacijo značilk LDA

na Sliki 3.1.

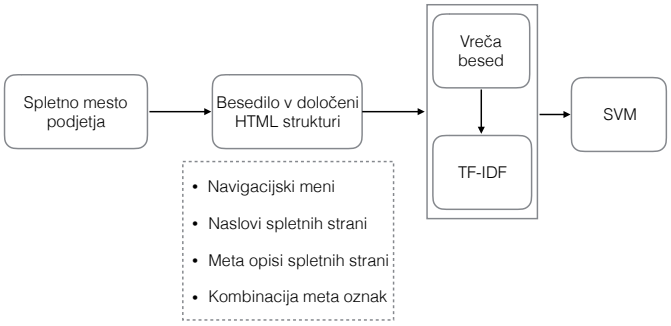
Seveda pri tem modelu za besedilo, ki ga potem analiziramo, ne vzamemo preprosto celotne spletne strani. Da iz spletne strani (in posledično iz spletnega mesta) dobimo besedilo, najprej izberemo le en določen del strani. Pri našem osnovnem modelu je to element HTML *body*. Pri alternativnih modelih se ta element razlikuje, s čimer želimo doseči čim boljše rezultate. Nato iz tega dela spletne strani besedilo dobimo tako, da odstranimo vse strukture jezika HTML. Ostane nam le besedilo, ki smo ga želeli izluščiti.

Na podlagi raziskave [16] smo se odločili preizkusiti še variacijo tega napovednega modela. V raziskavi so pri klasifikaciji tekstov dosegli izboljšavo rezultatov, ko so značilkam (vreča besed) dodali še izhodni vektor algoritma LDA. Za variacijo našega napovednega modela smo tako preizkusili še dodatne značilke, ki nam jih vrne algoritem LDA. Na Sliki 3.2 lahko vidimo diagram te variacije modela.
Prav tako smo na podlagi raziskave [8] preizkusili, če lahko dosežemo boljše rezultate z uporabo analize LSA. Diagram modela lahko vidimo na Sliki 3.3.

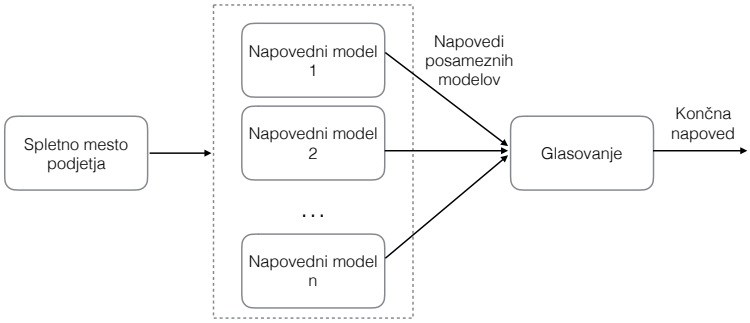
Želeli smo preveriti tudi uporabo alternativnih značilk pri napovedovanju



Slika 3.3: Diagram delovanja napovednega modela z uporabo LSA



Slika 3.4: Diagram delovanja alternativnih napovednih modelov



Slika 3.5: Diagram delovanja združevalnega napovednega modela

panoge. Ker nam jezik HTML podaja neko strukturo besedila, jo je smiselno izkoristiti kot dodaten vir informacij. Nekaj struktur, ki se izkažejo za zelo uporabne [7, 8, 12, 1], smo se odločili preizkusiti:

- navigacijski meni spletišča,
- naslovi (title elementi) spletnih strani spletišča,
- meta opisi spletnih strani spletišča,
- meta ključne besede spletnih strani spletišča in
- kombinacija meta oznak (naslovi, opisi in ključne besede).

S tem smo dobili še 5 alternativnih napovednih modelov, ki so bili v strukturi podobni osnovnemu. Namesto da smo vrečo besed izvedli nad celotnim spletiščem, smo jo izvedli samo nad posamezno strukturo, ki smo jo v danem modelu želeli preizkusiti. Ker smo s tem imeli skupaj 6 različnih napovednih modelov, smo se za konec odločili preveriti še napovedni model, ki združuje vse ostale modele. S tem smo želeli dobiti boljše rezultate, kot

nam jih je vrnil le posamezni model. Diagram delovanja posameznih alternativnih modelov lahko vidimo na Sliki 3.4 in diagram združevalnega modela na Sliki 3.5.

3.1.1 Implementacija

Koda 1 Implementacija pridobivanja celotnega teksta spletnega mesta

```
1  from bs4 import BeautifulSoup
2
3  def get_full_text(website):
4      text = ''
5      # Preglej vse spletne strani posameznega spletnega mesta
6      for webpage_path in webpages_in_website(website):
7          # Pridobi html strukturo spletne strani
8          with open(webpage_path, 'r') as f:
9              html = BeautifulSoup(f, 'html.parser')
10         # Shrani celotno besedilo, ki je najdeno znotraj
11         # elementa <body>
12         body = html.body
13         if body:
14             # Vendar brez besedila najdenega znotraj <script>
15             for script in body.find_all('script'):
16                 script.clear()
17             text += body.get_text()
18             text += '\n'
19     return text
```

V izseku kode 1 lahko vidimo implementacijo pridobivanja besedila celotnega spletnega mesta in vračanje tega besedila kot en dokument in ne kot skupek dokumentov. Tu nam močno pomaga *Beautiful Soup 4*, saj nam omogoča, da z ukazom `html.body.get_text()` dobimo vso besedilo znotraj

elementa *body* kot besedilo in ne kot HTML. Opazimo lahko tudi, da pred pridobivanjem besedila, v vrstici 16 spletno stran očistimo vseh *script* elementov. To storimo zato, ker bi sicer z ukazom `.get_text()` izluščili tudi vsebino teh elementov. Izkaže se, da bi brez odstranitve teh elementov, v naše modele vnesli visoko stopnjo šuma, kar seveda ni zaželjeno. Vse skupaj je verjetno posledica tega, da te elementi vsebujejo JavaScript kodo in ne besedila.

Na podoben način smo iz spletnih mest izluščili tudi druge dele. V izseku kode 3 je podana implementacija pridobivanja naslovov spletnih strani (iz *title* elementov jezika HTML). Podobno v izseku kode 2 lahko vidimo implementacijo pridobivanja opisov spletnih strani in pridobivanje ključnih besed spletne strani. Pri pridobivanju vseh meta oznak lahko opazimo, da moramo preveriti, če iskani element (*title* ali *meta* z atributom *description* ali z atributom *keywords*) na spletni strani sploh obstaja. Element *title* je sicer glede na HTML4 in HTML5 standard obvezen[2], vendar se, tako kot je pogosto tudi pri drugih elementih, spletne strani tega ne držijo vedno. V primeru, da elementa ne najdemo, nam ne preostane nič drugega, kot da preprosto nadaljujemo iskanje po drugih straneh posameznega spletišča.

Malce bolj pa je zapleteno pridobivanje navigacijskih menijov iz posameznih spletnih strani. HTML5 standard je sicer uvedel *nav* element[3], vendar vse spletne strani ne sledijo HTML5 standardu. Seveda pa tudi za tiste, ki standardu sledijo, ni nobenega zagotovila, da bodo *nav* zares uporabile kot navigacijski meni. Posledično je bila implementacija pridobivanja navigacijskih menijev veliko bolj zapletena. Poskušali smo pokriti najbolj pogoste implementacije navigacijskih menijev na spletnih straneh. Vendar ker je različnih možnost neomejeno, za veliko spletnih mest podjetij nismo uspeli najti navigacijskega menija. Implementacija, ki jo vidimo v izseku kode 4, je bila razvita empirično; večkrat smo dodali nove korake na podlagi spletnih strani, ki smo jih imeli na voljo, da bi algoritem ujel čim več različnih

[2]https://html.spec.whatwg.org/multipage/dom.html#the-title-attribute
[3]https://html.spec.whatwg.org/multipage/semantics.html#the-nav-element

možnosti navigacijskih menijev. Na neki točki smo smatrali algoritem za zadosti učinkovit za naše potrebe. Naredili smo tudi predpostavko, da imajo vse spletne strani istega spletnega mesta isti navigacijski meni.

Kot nam narekuje naš napovedni model, po pridobitvi različnih besedil nad njimi izvedemo vrečo besed in za značilke za klasifikator vzamemo vrednosti tf-idf. Kot lahko vidimo v izseku kode 5, smo za implementacijo uporabili `HashingVectorizer` in `TfidfTransformer` iz programskega paketa *scikit-learn*. Za lažji razvoj in boljšo preglednost smo si pomagali tudi z `Pipeline`, ki nam omogoča, da zaporedno zvrstimo različne pretvorbe vhodnih podatkov. Na ta način tudi zagotavljamo lažje primerjanje rezultatov in zmanjšamo možnost napak pri testiranju različnih variacij. Za izračun vreče besed *scikit-learn* ponuja tudi `CountVectorizer`, ki pa ga nismo uporabili zaradi počasnejše in manj efektivne implementacije. `HashingVectorizer` je namreč optimiziran za velike, redke matrice, kar je pri količini podatkov, s katerimi smo rokovali, zelo pomembno. Za SVM klasifikator, smo uporabili implementacijo `SGDClassifier`, ki za učenje uporablja metodo SGD.

Pri tem modelu smo morali določiti več parametrov. Za `HashingVectorizer` smo določili sledeče parametre:

- `input` – določa, kako bo algoritem prebral vhodni podatek. Uporabili smo `'filename'`, saj je pomembno, da sistem bere direktno iz shranjenih datotek in ne da besedila nalagamo v delovni spomin.

- `ngram_range` – določa, ali naj vreča besed upošteva unigrame – `(1, 1)`, bigrame – `(1, 2)`, ali n-grame – `(1, n)`. Zaradi velikih dimenzij vektorjev značilk ob uporabi bi- ali n-gramov smo za osnovni model uporabili unigrame. Preverili smo tudi uspešnost napovedovanja pri uporabi bi- ali n-gramov.

- `stop_words` – določa, ali naj iz besedila odstrani odvečne besede (angl. *stop words*). Za osnoven model se za odstranjevanje odvečnih besed nismo odločili – `stop_words=None`. Pri preizkušanju različnih parametrov pa smo preizkusili odstranjevanje odvečnih besed v angleščini –

`stop_words='english'`.

Pri algoritmu `SGDClassifier` smo s parametrom `loss='hinge'` določili uporabo SVM algoritma (namesto kakšne druge možnosti, recimo logistična regresija). Parametre `penalty`, `n_iter` in `alpha` smo optimizirali s pomočjo prečnega preverjanja znotraj naše učne množice. V modelu smo uporabili optimalne parametre, ki so prikazani v implementaciji. Pomemben je bil tudi parameter `n_jobs=10`, s katerim smo vključili vzporedno procesiranje naših podatkov. Ker je bila količina podatkov velika, so bile vse možnosti za hitrejše računanje zelo dobrodošle.

Za variacijo našega osnovnega modela, ki za značilke izkorišča tudi izhod algoritma LDA, je bila potrebna malce prirejena implementacija, ki jo lahko vidimo v izseku kode 6. Za izračun LDA smo uporabili `LatentDirichlet-Allocation`. S predhodnim preverjanjem smo ugotovili, da se izbira parametra `n_topics=100` obnese najbolje. Ta nam določa število izhodnih značilk algoritma LDA. Zaradi računske zahtevnosti izračuna smo morali omejiti število značilk, ki nam jih vrne vreča besed skupaj z tf-idf. Ker nam `HashingVectorizer` tega ne omogoča, smo uporabili `TfidfVectorizer`, ki v ozadju uporablja `CountVectorizer` skupaj z `TfidfTransformer`. Število značilk smo omejili s parametrom `max_features=3000`. Vse parametre prikazane v implementaciji, smo izbrali na podlagi prečnega preverjanja znotraj učne množice. Vektorja tf-idf in izhod algoritma LDA smo združili s pomočjo `FeatureUnion`, ki vzame nespremenjene vhodne podatke (torej tf-idf) in izhod LDA ter jih združi.
Malce bolj preprosta je bila implementacija modela, ki značilke izračuna z analizo LSA. Za implementacijo smo uporabili `TruncatedSVD`, ki pri vhodu tf-idf vrednosti vrne izračun LSA. Implementacijo lahko vidimo v izseku kode 7.

Postopek opisan v izseku kode 5 smo uporabili za vse različne vire tekstov, ki smo jih opisali zgoraj. S tem smo dobili 6 različnih napovedi in da bi izboljšali rezultat posameznih napovedi, smo za zadnji model združili ostalih 6 modelov, kot je razvidno v diagramu na Sliki 3.5. V izseku kode 8 lahko vi-

dimo, kako smo s pomočjo `Pipeline` in `VotingClassifier` programskega paketa *scikit-learn* implementirali združevalni napovedni model. Za združiteve rezultatov smo uporabili preprosto tehniko glasovanja. Posamezni napovedni modeli pa sledijo modelu opisanem v izseku kode 5, le da vsak uporabi različne vhodne podatke (celotno besedilo, le navigacijske menije, itd.) in vsak je naučen s svojim klasifikatorjem. V `VotingClassifier` smo za parameter `voting`, ki določa način glasovanja, uporabili `'soft'`. Ta za napoved upošteva verjetnosti napovedi, ki bi nam jih podal vsak posamezen model. Parameter `weigths` nam določa do kakšne mere naj algoritem upošteva posamezno napoved. S tem parametrom lahko kontroliramo, kateri modeli želimo, da pri končni napovedi igrajo večjo vlogo. Na primer z nastavitvijo parametra na `weigths=[1, 1, 1, 1, 1, 1]`, se vse napovedi upošteva enakovredno. Medtem ko z nastavitvijo parametra na `weigths=[1, 1, 0, 0, 0, 0]` upoštevamo le prvo in drugo napoved. Uporabimo lahko tudi vmesne vrednosti, recimo `weigths=[1.5, 1, 1, 1, 1, 1]`.

3.1.2 Metrika uspešnosti

Za uspešno določitev najboljšega modela in ustreznih parametrov je ključno tudi definirati, kaj za nas pomeni "najboljši model". Metrika uspešnosti mora odražati, kaj želimo z našim modelom doseči. Za naš model smo se odločili uporabiti metriko priklica, saj nam bo ta metrika dobro odražala učinkovitost in pokritost. Ker imamo opravka z večrazredno klasifikacijo, moramo priklic med vsemi panogami tudi nekako povprečiti. Ker so naši podatki neuravnoteženi, bomo priklic povprečili uteženo glede na zastopanost posamezne panoge, čemur ponavadi rečemo uteženo povprečenje (angl. *weighted averaging*).

Obenem je pri izbiri metrike uspešnosti pomembno tudi upoštevanje podobnosti posameznih panog. V Poglavju 2.3 smo govorili o problemu subjektivnosti pri panogah. Za primer lahko pogledamo izsek seznama panog v Tabeli 3.1. Opazimo, da so si nekatere panoge precej bolj podobne kot druge. To smo morali upoštevati tudi pri ocenjevanju uspešnosti našega mo-

Tabela 3.1: Izsek iz tabele panog

Panoga
...
Building Materials
...
Computer Software
Computer Games
Cosmetics
...
Information Services
Information Technology and Services
Internet
...

dela. Saj na primer če za neko podjetje, ki ima panogo *Computer Games*, napačno napovemo panogo *Computer Software* ali *Internet*, je ta napaka mnogo manjša, kot če bi napačno napovedali recimo *Cosmetics* ali *Building Materials*. Zato bi lahko rekli, da je napaka *Computer Software* namesto *Computer Games* samo delna napaka in tako napako zato kaznujemo manj kot kakšno drugo napako. To smo upoštevali tako, da smo vsaki panogi najprej pripisali seznam panog, ki so tej panogi zadosti podobne. Nato smo pri računanju priklica vsake napačne napovedi, ki je znotraj *podobnih* panog, pripisali vrednost 0.5, namesto 0. Implementacijo tega lahko vidimo v izseku kode 9. Seznam *podobnih* panog je na voljo v prilogi A.

V teoriji je najboljši možen rezultat priklica 1, v praksi pa tudi priklic klasifikacije, ki bi jo izvedel človek, ne bi presegla vrednosti 0.8 ali 0.9. To je posledica subjektivnosti teh napovedi in težke določljivosti nekaterih primerov.

3.2 Število zaposlenih

Za napovedovanje števila zaposlenih v podjetju smo želeli uporabiti drugačne modele oziroma značilke, kot smo jih uporabili za napoved panoge. To pa zato, ker predpostavljamo, da besedilna vsebina spletnih strani nima direktne povezave z velikostjo podjetja. Zato rajši preizkusimo različne meta podatke, ki jih uspemo razbrati iz spletnih mest podjetij. Na Sliki 3.6 vidimo diagram modela za napovedovanje števila zaposlenih. Opazimo, da za razliko od modela za napovedovanje panoge, tu ne uporabljamo tehnik NLP, zaradi česar je ta model računsko mnogo manj zahteven.

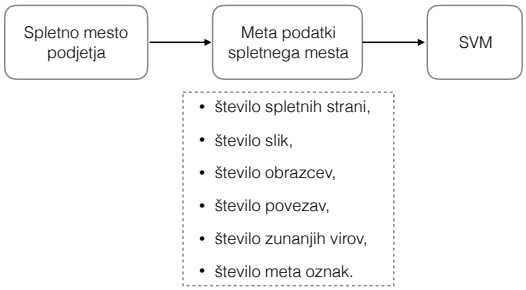
Značilke, ki smo jih v tem modelu želeli uporabiti, so različni meta podatki spletnega mesta podjetja in sicer:

- število spletnih strani,
- število slik oziroma število *img* elementov,
- število obrazcev oziroma število *form* elementov,
- število povezav oziroma število *a* elementov,
- število zunanjih virov oziroma *link* elementov in
- število meta oznak oziroma število *meta* elementov.

Te značilke smo izbrali na podlagi različnih možnih značilk za opisovanje spletnih mest, ki sta jih v raziskavi opisala Bauer, Christian in Scharl, Arno [17]. Te značilke lahko na preprost način dobimo iz katere koli spletne strani in omogočajo nam, da se izognemo računsko zahtevni operaciji obdelave besedil.

3.2.1 Implementacija

Implementacija pridobivanja teh značilk je bila s pomočjo *Beautiful Soup 4* trivialna, kot lahko vidimo v izseku kode 10. Potrebno pa je omeniti, da se



Slika 3.6: Diagram delovanja modela za napovedovanje števila zaposlenih

tu na nek način zanašamo na podatke, za katere ne moremo zagotavljati, da odražajo realno stanje. Ker ne moremo vedeti ali spletno mesto v naših podatkih vsebuje zares vse spletne strani tega spletnega mesta, pomeni, da ne moremo z zagotovostjo trditi, da so razbrani meta podatki točni. Ker pa smo za vse vzorce v naboru podatkov uporabili isti način pridobivanja spletnega mesta, predpostavljamo, da to ne bo predstavljalo problema.

Podobno kot pri napovedi panoge smo tudi tu za model izkoristili `Pipeline` in `SGDClassifier`. Implementacijo lahko vidimo v izseku kode 11.

3.2.2 Metrika uspešnosti

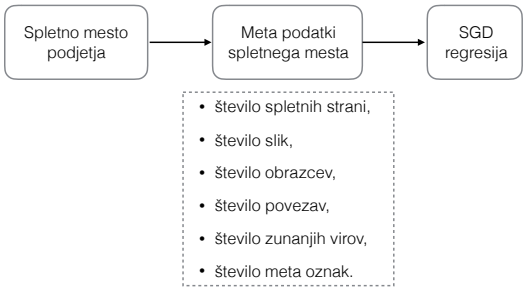
Pri napovedovanju števila zaposlenih smo za metriko uspešnosti uporabili klasifikacijsko točnost. Vendar smo tudi to metriko, podobno kot pri napovedovanju panoge, primerno modificirali glede na naš primer. Točna napoved števila zaposlenih je namreč zelo stroga metrika. Želeli smo upoštevati dejstvo, da je napoved, ki se zmoti le za en razred (recimo napoved 1-10 zaposlenih namesto dejanskih 11-50 zaposlenih), za nas še vedno sprejemljivo. To

smo upoštevali tako, da smo pri izračunu točnosti za napačno napoved vzeli le primere, ko se napoved razlikuje za več kot 1 razred, glede na razporeditev omenjeno v Poglavju 2.5.

Poleg osnovne napovedi pa smo želeli preveriti še, če bi lahko dobili boljše rezultate, če bi ta problem obravnavali kot regresijo namesto klasifikacijo. Ker so naši podatki o številu zaposlenih razvrščeni v 9 različnih kategorij, smo zvezne podatke morali ustvariti iz obstoječih diskretnih podatkov. To smo storili preprosto tako, da smo vsakemu intervalu pripisali srednjo vrednost in potem to vrednost upoštevali kot število zaposlenih. Za metriko uspešnosti smo upoštevali povprečno absolutno napako. Da bi lahko primerjali rezultate s klasifikacijo, smo končne napovedi umestili še v iste intervale, kot smo jih uporabili na vhodu. Nato pa nad temi prirejenimi napovedmi izračunali točnost na isti način, kot pri klasifikaciji.

3.3 Starost

Podobno kot pri napovedovanju števila zaposlenih smo tudi pri napovedovanju starosti podjetja želeli izkoristiti druge značilke kot samo vsebino spletnega mesta podjetja. Za starosti podjetja predpostavljamo, da ima še manj povezave ne le z besedilno vsebino spletnega mesta temveč tudi s samo spletno stranjo. Razlog za to je, da nam že sam podatek o starosti podjetja velikokrat ne pove veliko. Kot leto ustanovitve podjetja navajajo marsikaj. Podjetja, ki imajo tradicijo, to zelo rada poudarjajo, kar pomeni, da jih veliko za leto ustanovitve poda prve začetke podjetja. Ta lahko v nekaterih primerih segajo ne le v 2. temveč tudi v 1. tisočletje. Po drugi strani pa lahko podjetje, ki deluje že vrsto let, po kakšni reorganizaciji navaja za leto ustanovitve leto reorganizacije. Ker pa se tudi pri tem podatku zanašamo na informacijo, ki so jo podala podjetja sama, ne moremo z nobeno gotovostjo trditi, kaj točno starost podjetja za nas pomeni.



Slika 3.7: Diagram delovanja modela za napovedovanje starosti

3.3.1 Metrika uspešnosti

Zaradi problematike podatkov smo želeli preizkusiti še alternativno napovedovanje, ki bi nam mogoče podalo zgolj tisto informacijo, ki nas zanima. Za primer smo vzeli napovedovanje ali je podjetje mlado ali ne. Zanimalo nas je, če problem obravnavamo kot klasifikacijski problem in zmanjšamo nabor možnosti na samo *mlado* (podjetje staro 10 let ali manj) in *staro* (podjetje starejše od 10 let) podjetje, ali lahko pridemo do bolj uporabnih rezultatov. Pri tem smo uporabili iste značilke, le za algoritem strojnega učenja smo izbrali regresijo, kot lahko vidimo na Sliki 3.7.

Za metriko uspešnosti smo v primeru regresije uporabili povprečno absolutno napako, v primeru klasifikacije pa natančnost in priklic napovedi za *mlada* podjetja.

Koda 2 Implementacija pridobivanja meta oznak spletnega mesta

```
from bs4 import BeautifulSoup

def get_meta_descriptions_text(website):
    text = ''
    prev_description = None
    # Preglej vse spletne strani posameznega spletnega mesta
    for webpage_path in webpages_in_website(website):
        # Pridobi html strukturo spletne strani
        with open(webpage_path, 'r') as f:
            html = BeautifulSoup(f, 'html.parser')
        # Najdi element, ki vsebuje opis spletne strani
        desc_el = html.find('meta', attrs={'name':'description'})
        if desc_el:
            desc = desc_el.get('content')
            # Preveri, da ne dodamo istega teksta,
            # ki smo ga ravnokar dodali
            if prev_description != desc:
                prev_description = desc
                text += desc
                text += '\n'
    return text

def get_meta_keywords_text(website):
    text = ''
    # Preglej vse spletne strani posameznega spletnega mesta
    for webpage_path in webpages_in_website(website):
        # Pridobi html strukturo spletne strani
        with open(webpage_path, 'r') as f:
            html = BeautifulSoup(f, 'html.parser')
        # Najdi element, ki vsebuje opis spletne strani
        keywords_el = html.find('meta', attrs={'name':'keywords'})
        if keywords_el:
            keywords = keywords_el.get('content')
            text = ' '.join(keywords)
            # Ko smo nasli kljucne besede
            # prekinimo iskanje
            break
    return text
```

Koda 3 Implementacija pridobivanja naslovov spletnega mesta

```
from bs4 import BeautifulSoup

def get_titles_text(website):
    text = ''
    # Preglej vse spletne strani posameznega spletnega mesta
    for webpage_path in webpages_in_website(website):
        # Pridobi html strukturo spletne strani
        with open(webpage_path, 'r') as f:
            html = BeautifulSoup(f, 'html.parser')
        # Najdi element, ki vsebuje naslov spletne strani
        title_el = html.find('title')
        if title_el:
            title = title.string
            text += title
            text += '\n'
    return text
```

Koda 4 Implementacija iskanja navigacijskega menija na spletni strani

```
from bs4 import BeautifulSoup

def get_nav_menu(html):
    # Navigacijski meni pricakujemo v glavi elementa body
    # Najprej poskusimo najti glavo
    header = html.body.find_all('header')
    if not header:
        header = html.body.find_all({'class': ['header', 'menu']})
    if not header:
        header = html.body.find_all(id='header')
    # Ce smo jo uspeli najti, jo uporabimo
    if header:
        header = header[0]
        has_header = True
    # Sicer rajsi iscemo drugje
    else:
        has_header = False
        header = html.body
    # Poskusi najti navigacijski meni na razlicnih mestih
    nav = header.find('nav')
    if not nav:
        nav = html.body.find('nav')
    if not nav:
        nav = header.find(class_='nav')
    if not nav:
        nav = header.find(id='nav')
    if not nav:
        nav = header.find(class_='menu')
    if not nav:
        nav = header.find(class_='main-menu')
    # Ce smo nasli kaksne rezultate, jih zdruzimo v seznam
    if nav:
        return [item for li in nav.find_all('li')
                     for item in li.stripped_strings]
    elif has_header:
        return [item for li in header.find_all('li')
                     for item in li.stripped_strings]
    else:
        return []
```

Koda 5 Postopek učenja na podlagi vektorja tf-idf, zgrajenega na besedilu spletišča

```
from sklearn.feature_extraction.text import HashingVectorizer,
                                            TfidfTransformer
from sklearn.linear_model import SGDClassifier
from sklearn.pipeline import Pipeline

hv = HashingVectorizer(input='filename',
                       ngram_range=(1, 1),
                       stop_words=None)
tf_transformer = TfidfTransformer()
clf = SGDClassifier(loss='hinge',
                    penalty='l2',
                    n_iter=10,
                    alpha=0.001,
                    n_jobs=10)

pipeline_elements = [('hv', hv),
                     ('tfid', tf_transformer),
                     ('SVM', clf)]
pipe = Pipeline(pipeline_elements)

pipe.fit(train_data, target)
```

Koda 6 Postopek učenja s kombinacijo značilk LDA, zgrajenega na besedilu spletišča

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.decomposition import LatentDirichletAllocation
from sklearn.linear_model import SGDClassifier
from sklearn.pipeline import Pipeline, FeatureUnion

tfv = TfidfVectorizer(input='filename', max_features=3000)
clf = SGDClassifier(loss='hinge',
                    penalty='l2',
                    n_iter=10,
                    alpha=0.001,
                    n_jobs=10)
lda = LatentDirichletAllocation(n_topics=100,
                                max_iter=5,
                                n_jobs=5)

# Zdruzi znacilke tf-idf in LDA
lda_union = FeatureUnion([('lda', lda),
                          ('identity', identity)
                         ])
pipe_lda = Pipeline([('tfv', tfv),
                     ('lda_union', lda_union),
                     ('clf', clf)
                    ])

pipe_lda.fit(train_data, target)
```

Koda 7 Postopek učenja na podlagi LSA

```
from sklearn.feature_extraction.text import HashingVectorizer,
                                            TfidfTransformer
from sklearn.decomposition import TruncatedSVD
from sklearn.linear_model import SGDClassifier
from sklearn.pipeline import Pipeline

hv = HashingVectorizer(input='filename',
                       ngram_range=(1, 1),
                       stop_words=None)
tf_transformer = TfidfTransformer()
lsa = TruncatedSVD(n_components=1000)
clf = SGDClassifier(loss='hinge',
                    penalty='l2',
                    n_iter=10,
                    alpha=0.001,
                    n_jobs=10)

pipeline_elements = [('hv', hv),
                     ('tfid', tf_transformer),
                     ('lsa', lsa)
                     ('SVM', clf)]
pipe = Pipeline(pipeline_elements)

pipe.fit(train_data, target)
```

Koda 8 Postopek učenja na podlagi glasovalnega modela

```
from sklearn.ensemble import VotingClassifier

voting = VotingClassifier(estimators=[('full', pipe_full),
                                      ('navs', pipe_nav_menus),
                                      ('desc', pipe_descriptions),
                                      ('titles', pipe_titles),
                                      ('keywords', pipe_keywords),
                                      ('combined_meta', pipe_meta)],
                          voting='soft',
                          weigths=[1, 1, 1, 1, 1, 1])

voting.fit(train_data, target)
```

Koda 9 Izračun uteženega priklica

```
from collections import Counter

def score_similar(estimator, data, actual):
    predict = estimator.predict(data)
    expected_count = Counter(actual)
    results = {}
    # Preverimo za vsak testni vzorec
    for num, example in enumerate(actual):
        # Ce se ujema
        if example == predict[num]:
            results[example] = results.get(example, 0) + 1
        # ali ce se delno ujema
        elif predict[num] in similar_industries.get(example, []):
            results[example] = results.get(example, 0) + 0.5

    # Izracunamo priklic
    for key, value in results.items():
        results[key] = value / expected_count[key]

    # Povprecimo glede na zastopanost
    return sum([v * expected_count[key] / len(actual)
                for key, v in results.items()])
```

Koda 10 Implementacija pridobivanja meta podatkov spletnega mesta

```
from bs4 import BeautifulSoup

def get_meta_features(website):
    image_count = 0
    sites = 0
    form_count = 0
    a_count = 0
    link_count = 0
    meta_count = 0
    # Preglej vse spletne strani posameznega spletnega mesta
    for webpage_path in webpages_in_website(website):
        # Pridobi html strukturo spletne strani
        with open(webpage_path, 'r') as f:
            html = BeautifulSoup(f, 'html.parser')
        # Pridobi zeljene meta podatke
        sites += 1
        image_count += len(html.find_all('img'))
        form_count += len(html.find_all('form'))
        a_count += len(html.find_all('a'))
        link_count += len(html.find_all('link'))
        meta_count += len(html.find_all('meta'))
    return [sites, image_count,
            form_count, link_count, meta_count]
```

Koda 11 Postopek učenja na podlagi meta podatkov o spletnem mestu

```
from sklearn.linear_model import SGDClassifier
from sklearn.pipeline import Pipeline

clf = SGDClassifier(loss='hinge',
                    penalty='l2',
                    n_iter=10,
                    alpha=0.001,
                    n_jobs=10)
pipeline_elements = [('get_meta', get_meta_features),
                     ('SVM', clf)]
pipe = Pipeline(pipeline_elements)

pipe.fit(train_data, target)
```

Poglavje 4

Rezultati

Rezultati predstavljeni v tem poglavju so bili izračunani na testni množici, ki je bila iz faze učenja modelov strogo izključena. Na ta način smo se izognili prekomernemu prileganju naših modelov, obenem pa zagotovili veljavnost rezultatov.

4.1 Panoga

Še preden opravimo prvo meritev oziroma primerjavo rezultatov, je smiselno najprej preveriti, kakšne metrike uspešnosti dobimo za napovedni model, ki nam vrača naključne ali konstantne rezultate. To nam da nek osnoven vpogled v uspešnost našega modela. Prav tako se lahko prepričamo, da nam naša metrika uteženega priklica ne daje neutemeljeno predobrih rezultatov. Pri preverjanju takih naključnih napovednih modelov nam je v veliko pomoč `DummyClassifier` iz paketa *scikit-learn*. Tak klasifikator sprejme argument `strategy`, s katerim lahko določimo, katero strategijo želimo preveriti. Za naključno napovedovanje panoge podjetja smo preizkusili strategije:

- `'stratified'`, ki vrača naključne rezultate, utežene glede na zastopanost posamezne kategorije,
- `'most_frequent'`, ki vedno vrne kategorijo, ki je najbolj zastopana in

Tabela 4.1: Uspešnost napovedovanja naključnega klasifikatorja

Strategija napovedovanja	Utežen priklic
`'stratified'`	4.2%
`'most_frequent'`	8.3%
`'uniform'`	2.4%

- `'uniform'`, ki vrača enotno naključne rezultate.

V Tabeli 4.1 lahko vidimo, da je najboljši rezultat, ki nam ga tak klasifikator vrne, 8.3%.

Napoved modela, ki je kot vhod upošteval besedilo celotnega spletišča, je bil naša osnova. Sklepali smo, da bomo iz vreče besed nad celotnim spletiščem uspeli dobiti neke smiselne rezultate, ki pa smo jih nato želeli izboljšati. Ta model je z optimiziranimi parametri dosegel utežen priklic 52.3%. Ta rezultat se na prvi pogled mogoče zdi slab, vendar je pri tem treba nujno upoštevati, da smo napovedovali med 90 različnimi panogami.

V Tabeli 4.2 lahko vidimo rezultate osnovnega modela in rezultate vseh alternativnih modelov, ki so se razlikovali v vhodu, ki smo ga uporabili za izračun vreče besed. Opazimo, da vsi dosežejo boljše rezultate kot naključen klasifikator, vendar slabše od osnovnega klasifikatorja. Še najbolj se mu približa klasifikator, ki za vhod uporabi kombinacijo vseh meta oznak in doseže utežen priklic 44.8%. Zanimiv je tudi rezultat modela, ki uporablja meta opise in doseže rezultat 31.4%. To je še posebej zanimivo, saj meta opise najdemo le v 60% vzorcev našega nabora podatkov.
Pri osnovnem modelu smo preizkusili tudi druge načine pridobivanja značilk. Med drugim smo preverili uporabo bigramov, ngramov in odstranitvijo odvečnih besed (angl. *stop words*). Nobena kombinacija ni bistveno izboljšala rezultata osnovnega modela, kot lahko vidimo v Tabeli 4.4. Zanemarljivo boljše sta se obnesla modela z bigrami in n-grami, a ker sta modelu bistveno povečala računsko zahtevnost, smo se odločili, da jih ne vključimo v končen

model.
Da bi dosegli boljši rezultat od rezultatov posameznih modelov, smo preizkusili tudi model, ki preko glasovanja združuje vse ostale modele. Nasprotno s pričakovanji, je ta model dosegel malce slabši rezultat kot le osnovni model in sicer 49.6%. Vendar ko smo kombinacijo modelov utežili s primernimi utežmi, smo dosegli željeno izboljšavo. Tak model je dosegel utežen priklic 54.6%. Pri tem smo prednost dajali osnovnemu modelu in kombinaciji meta oznak. Kot najbolj učinkovita kombinacija se je izkazala tista, pri kateri smo upoštevali le te dve napovedi in ostale izpustili. To pomeni, da smo pri implementaciji uporabili parameter `weigths=[1, 0, 0, 0, 0, 1]`.

Tabela 4.2: Rezultati napovednih modelov za napovedovanje panoge

Vhodni podatki	Utežen priklic
Celotno spletišče	52.3%
Celotno spletišče + LDA	36.5%
Celotno spletišče + LSA	44.7%
Navigacijski meni	29.9%
Meta opisi	31.4%
Naslovi	40.2%
Ključne besede	30.0%
Meta oznake	44.8%
Kombinacija	49.6%
Utežena kombinacija	54.6%

Ker vhodni podatki (spletna mesta podjetij) za naše napovedi živijo na spletu, smo želeli, da bi se modeli dobro odrezali ne glede na jezik spletnega mesta. Na ta način bi dosegli tudi čim bolj splošno rešitev. Nabor podatkov, ki smo ga uporabili, je vseboval spletna mesta različnih jezikov. Rezultati v Tabeli 4.2 so torej že rezultati, ko obravnavamo spletna mesta različnih

jezikov. Da bi primerjali te rezultate s primerom, ko imamo opravka le z enim jezikom, smo nabor podatkov zmanjšali na množico, ki vsebuje le angleška spletna mesta. Za določanje jezika smo uporabili knjižnjico *langdetect* [1]. Nato smo na isti način, le z uporabo te nove množice podatkov, znova naučili in testirali naše modele. Rezultati, ki jih lahko vidimo v Tabeli 4.3, so se sicer izboljšali, a ne bistveno. Utežen združevalni klasifikator je recimo dosegel utežen priklic 55.2%. S tem smo pokazali, da se model obnese dobro, tudi ko jezik ni enoten.

Tabela 4.3: Rezultati napovednih modelov za napovedovanje panoge v primeru enotnega jezika

Vhodni podatki	Utežen priklic
Celotno spletišče	54.9%
Celotno spletišče + LDA	42.2%
Celotno spletišče + LSA	48.8%
Navigacijski meni	30.3%
Meta opisi	32.9%
Naslovi	42.0%
Ključne besede	30.8%
Meta oznake	46.6%
Kombinacija	49.9%
Utežena kombinacija	55.2%

V prejšnjih rezultatih smo se osredotočali na uspešnost modela kot celoto. Uporabnost metrike povprečnega uteženega priklica je v primerjavi med posameznimi modeli. Ko pa smo našli optimalen model, nas je seveda zanimalo bolj podrobno, kako se odreže v napovedovanju specifične panoge. Zato smo si ogledali utežene priklice za vsako panogo posebej. V Tabeli 4.5 je prikazanih nekaj panog pri katerih se je model še posebej dobro obnesel. Opazimo,

[1] https://github.com/Mimino666/langdetect

Tabela 4.4: Rezultati uporabe različnih parametrov modela za napovedovanje panoge

Model	Utežen priklic
Osnovni model	52.3%
Osnovni model + bigrami	52.6%
Osnovni model + n-grami	52.5%
Osnovni model + stop words	52.1%

da model panogo *zavarovalništvo* napoveduje z 87.7% uteženim priklicom, kar je precej boljše kot povprečje 54.6%.
V Tabeli 4.6 pa vidimo nekaj primerov, pri katerih model napoveduje mnogo slabše od povprečja. V nekaterih primerih tudi slabše od 15%.
V Prilogi B lahko vidimo rezultate še za vse ostale panoge.

4.2 Število zaposlenih

Enako kot pri modelu za napovedovanje panoge smo tudi tu najprej želeli preizkusiti, kako se obnese naključen napovedni model. Rezultate točnosti in utežene točnosti takega modela lahko vidimo v Tabeli 4.7. Vrednosti so zaradi utežene točnosti, ki je v tem primeru precej bolj blaga metrika, razmeroma visoke.

Naš klasifikacijski napovedni model je pri napovedovanju števila zaposlenih dosegel uteženo točnost 53.2%. Ko smo problem obravnavali kot regresijo, smo dosegli povprečno absolutno napako 393.8. To napoved smo tudi umestili v iste kategorije, ki jih je napovedoval prvi model, da bi lahko primerjali oba rezultata. V tem primeru se je utežena točnost povišala na 64.9%. V Tabeli 4.8 lahko vidimo uspešnost napovedi glede na posamezno kategorijo števila zaposlenih.

Tabela 4.5: Utežen priklic glede na panogo — uspešna napoved

Panoga	Utežen priklic	Število vzorcev
Insurance	87.7%	1909
Law Practice	83.1%	1322
Higher Education	80.9%	2717
Cosmetics	79.1%	593
Marketing and Advertising	79.0%	13181
Hospitality	78.0%	3645
Staffing and Recruiting	77.8%	2855
Wine and Spirits	77.8%	549
Real Estate	77.5%	3213
Leisure, Travel & Tourism	76.7%	4709
Architecture & Planning	75.9%	2398
Automotive	72.7%	2343
Telecommunications	72.6%	2643
Banking	72.2%	1042
Music	70.0%	996
Information Technology and Services	69.4%	12388
Hospital & Health Care	69.3%	2572
Accounting	69.2%	1060
Financial Services	68.1%	4134
Sports	66.3%	1691
...	...	...

Tabela 4.6: Utežen priklic glede na panogo — neuspešna napoved

Panoga	Utežen priklic	Št. vzorcev
...	...	...
Internet	26.1%	6760
Management Consulting	26.0%	4861
Consumer Goods	25.4%	1371
International Trade and Development	25.0%	816
Graphic Design	18.9%	960
Media Production	18.8%	1881
Facilities Services	18.2%	725
Consumer Services	16.2%	735
Business Supplies and Equipment	14.1%	764
Wholesale	12.7%	1173
Individual & Family Services	11.1%	560
Online Media	6.1%	1055

Tabela 4.7: Uspešnost napovedovanja naključnega klasifikatorja

Strategija napovedovanja	Utežen priklic
`'stratified'`	22.7%
`'most_frequent'`	31.6%
`'uniform'`	11.8%

Tabela 4.8: Uspešnost napovedovanja št. zaposlenih v odvisnosti od št. zaposlenih

Št. zaposlenih	Utežena točnost
1	20.83%
1 - 10	44.60%
11 - 50	93.09%
51 - 200	81.78%
201 - 500	53.79%
501 - 1000	11.08%
1001 - 5000	11.30%
5001 - 10,000	9.18%
10,001+	0.0%

4.3 Starost

Naključni model nam za napoved starosti v primeru regresije vrača povprečno absolutno napako 25 let. Za regresijo smo izračun pridobili s pomočjo `DummyRegressor` iz paketa *scikit-learn*. Tak model vedno napove povprečno vrednost učne množice. V primeru klasifikacije pa smo vrednosti navedli v Tabeli 4.9

Naš model, ki je starost podjetja napovedoval na podlagi meta značilk

Tabela 4.9: Uspešnost napovedovanja naključnega klasifikatorja

Strategija napovedovanja	Točnost	Priklic	F1
`'stratified'`	31.4%	31.5%	31.5%
`'most_frequent'`	0%	0%	0%
`'uniform'`	50.4%	31.5%	38.8%
vedno *mlado*	100%	31.5%	47.9%

spletnega mesta, je dosegel najmanjšo povprečno absolutno napako 21.38 let. Zanimala nas je tudi klasifikacijska napoved, kjer podjetja razdelimo na *mlada* (10 let ali mlajša) in *stara* (starejša od 10 let). Pri tem je naš klasifikacijski model dosegel točnost 41.7% in priklic mladih podjetih 72.4%, torej F1 vrednost 52.9%.

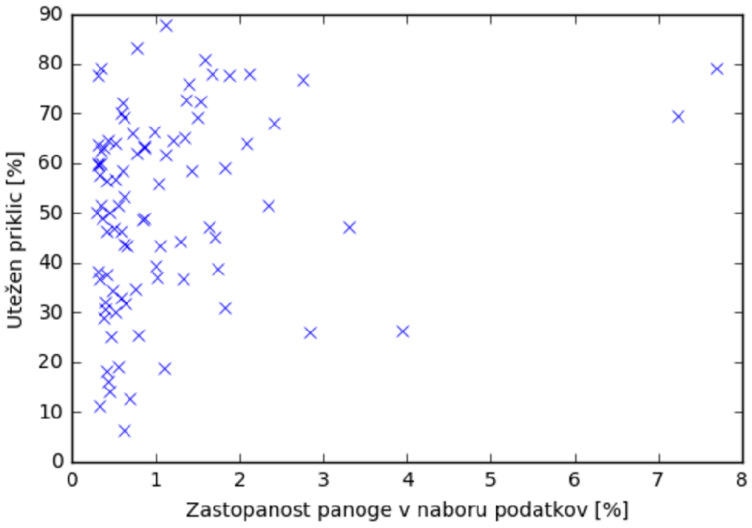
4.4 Sklepne ugotovitve

Za napovedovanje lastnosti podjetja smo preizkusili več različnih modelov in parametrov. Pri napovedovanju panoge smo dosegli najboljše rezultate z združevalnim modelom, ko smo ustrezno priredili uteži posameznih modelov. V nasprotju s pričakovanji se je neutežen združevalni model odrezal slabše kot le osnovni, kot lahko vidimo v Tabel 4.2. To je verjetno posledica tega, da so se vsi alternativni modeli odrezali občutno slabše od osnovnega. Alternativni modeli pa so se odrezali slabše deloma tudi zaradi dejstva, da smo potrebne značilke uspeli pridobiti le iz nekaterih spletnih mest. Prav tako nismo mogli doseči nobenih izboljšav z različnimi variacijami osnovnega modela predstavljenih v Tabeli 4.4. Predvidevamo, da ker že osnovni model ni dosegel boljših rezultatov, tudi uporaba različnih variacij ni pripomogla k izboljšavi, saj so bili glavni problemi za obstoječo napoved verjetno drugje. V Tabeli 4.5 in Tabeli 4.6 smo opazili, da se uspešnost naše napovedi precej razlikuje glede na panogo. Za specifične panoge smo dosegli veliko bolj zadovoljive rezultate. Obenem smo opazili tudi, da je večina panog za katere je bila naša napoved slaba, imela nižjo zastopanost v našem naboru podatkov, kot pa panoge za katere je bila napoved pravilnejša. Zaradi tega bi morda sklepali, da je slabša napoved posledica tega koliko vzorcev smo imeli za posamezno panogo. Vendar če primerjamo uspešnost napovedi v odvisnosti od zastopanosti, kot lahko vidimo na grafu na Sliki 4.1, očitne korelacije ne opazimo. Bolj zanimivo je opaženje, da so slabo napovedane panoge tiste, ki lahko vsebujejo veliko raznolikost. Recimo vzorcev s panogo *Internet* je sicer 6760 pa vendar je utežen priklic le 26.1%. Sklepamo, da je to posledica

tega, da je ta panoga zelo splošna in dovoljuje več različnih interpretacij. Možno je, da so to panogo navajala podjetja, ki se niso znala ali želela bolj specifično umestiti v drugo panogo. Posledično je pod isto panogo zastopano ogromno različnih podzvrsti, ki pa jih seveda ne zaznamo. Na drugi strani lahko vzamemo primer panoge *Cosmetics*, ki ima sicer zastopanost le 593, pa vendar je uspešnost napovedi med najvišjimi. V tem primeru je panoga zelo specifična in znotraj nje ne dovoljuje veliko raznolikosti.

Primerjava dobljenih rezultatov za napovedovanje panoge z rezultati iz sorodne raziskave [1] je težavna, saj smo pri delu uporabili popolnoma različen nabor podjetij, različne podatke in različne razvrstitve panog. Kljub temu lahko opazimo nekaj podobnosti in nekaj razlik. V raziskavi so najboljše rezultate dobili z uporabo besedil iz meta oznak, medtem ko smo v našem delu dobili najboljše rezultate s kombinacijo meta oznak ter celotnega besedila. Na prvi pogled se zdi, da smo pokazali ravno obratno a temu ni tako. Podobno kot smo opaili v tem delu, so tudi v raziskavi ugotovili, da meta oznake niso prisotne na vsakem spletnem mestu. Zato so vsakič, ko niso našli meta oznak, uporabili kar celotno besedilo. Kar pa je podobno temu, kar smo storili mi z združevalnim modelom. Torej prišli smo do istega zaključka, da je za klasifikacijo najbolj uporabna kombinacija besedila iz meta oznak in celotnega besedila.
V raziskavi so dosegli povprečen priklic 75%, kar je občutno boljše od naših rezultatov. Predvidevamo, da je to posledica manjše množice uporabljenih panog ter bolj fokusiranega nabora podatkov.
Podobno kot v našem delu, so tudi v raziskavi opazili, da se uspešnost napovedi za posamezno panogo precej razlikuje.

Pri napovedovanju števila zaposlenih smo dosegli presenetljivo dober rezultat glede na uporabljene značilke. Možno je, da bi dosegli še boljše rezultate, če bi uporabili bolj točne podatke. Razpolagali smo le s podatkom o tem, v kater interval števila zaposlenih nekatero podjetje spada. Boljše rezultate smo dosegli s tem, ko smo problem obravnavali kot regresijo. Glede na to, da smo za to morali umetno ustvariti zvezne podatke, sklepamo, da



Slika 4.1: Graf uspešnosti napovedi glede na zastopanost panoge

bi z dejanskimi vrednostmi lahko dosegli še boljši rezultat.

Model napovedovanja starosti podjetja se je na nek način izkazal za najslabšega, kar smo do neke mere pričakovali. Tu je bila problematična predvsem preslaba definicija tega, kaj želimo iz take napovedi zares izluščiti. Opazili smo, da podjetja za starost navajajo marsikaj in da je lahko razpon teh možnosti ogromen. Za vsako podjetje pa podatek o starosti pomeni tisto, kar sami želijo predstaviti o sebi. Nekatera podjetja bi rada poudarila svojo tradicijo, medtem ko bi se spet druga rada predstavila kot mlada podjetja. Način kako je podatek podan, ne preprečuje ne enega ne drugega. Posledica pa je težavna napoved te lastnosti.

Ob razvijanju napovednih modelov smo se srečali tudi z različnimi problemi, ki so bili v večini povezani le s podatki. Posvetiti smo morali veliko pozornosti čiščenju podatkov, da smo lahko imeli zadostno zaupanje v naše vhodne podatke. Kljub vsemu pa smo se zanašali na podatke s spleta, zaradi česar ni nobenih zagotovil, da bi podatki sledili določenim pravilom, katere bi želeli vzeti za privzeto. A količina podatkov je bila prevelika, da bi lahko ročno preverjali vse vzorce in s tem dosegli večjo zaupanje v podatke. Ta pro-

blem se je odražal tako na spletnih mestih, kot tudi na vseh lastnostih, ki smo jih želeli napovedati. Znotraj modela in z izbiro ustreznih metrik uspešnosti smo ta problem sicer precej uspešno reševali, vendar bi bilo za izboljšavo rezultatov in za nadaljne raziskave nujno potrebno rešiti ta problem na nivoju podatkov samih. Potrebno bi bilo posvetiti mnogo več časa pravilnemu shranjevanju spletnih mest in na različne načine preverjati njihovo veljavnost in celovitost. Tudi za podatke o lastnostih podjetja bi bilo potrebno bolje poskrbeti; idealno bi bilo, če bi lahko črpali iz kakšnega standardiziranega vira podatkov. Ker so taki viri ponavadi plačljivi in omejeni na posamezno državo, jih žal pri magistrskem delu nismo mogli uporabiti. Rezultate bi verjetno lahko tudi izboljšali z bolj fokusirano definiranim problemom. V tem delu smo napovedovali lastnosti podjetij iz različnih držav, povsem raznolikih panog, velikosti in tipov. Na primer model, ki bi se fokusiral le na eno državo, bi verjetno lahko dosegel boljše rezultate.

Navsezadnje pa glede na to, da smo napovedovali lastnosti, ki so le posredno povezane s spletnim mestom podjetja, včasih pa še to ne, so dobljeni rezultati realni. Za občutno boljše rezultate bi bilo verjetno potrebno uporabiti še kakšne druge podatke, kot le spletno mesto podjetja.

Poglavje 5

Zaključek

V našem delu smo se osredotočili na avtomatsko napovedovanje lastnosti podjetja na podlagi njihovega spletnega mesta. Napovedovali smo lastnosti kot so panoga, število zaposlenih in starost podjetja. Tak model, ki bi uspešno napovedoval te lastnosti, bi lahko v poslovnem svetu predstavljal konkurenčno prednost, saj so taki podatki velikokrat osnova za iskanje novih strank in sodelovanj. Najprej smo si v Poglavju 2 natančno pogledali nabor podatkov, ki je bil osnova za vse naše napovedne modele. Ker obstoječega nabora podatkov, ki bi bil primeren za naše delo, nismo našli, smo podatke najprej morali zbrati. Podatke smo pridobili iz prosto dostopnih virov na internetu in so temeljili na informacijah, ki jih podjetja poročajo sama. Temu primerno so bili podatki neusklajeni in v nekaterih primerih tudi neveljavni. Zato sta bila obdelava in čiščenje podatkov zelo pomembna koraka v celotnem postopku. Pri tem smo morali poskrbeti za:

- pridobitev veljavnega spletnega mesta podjetja,
- primeren seznam panog,
- uskladiti način podajanja informacije o številu zaposlenih in
- odstraniti vzorce z neveljavnimi podatki.

Za tem smo v Poglavju 3 natančno opisali predlagane napovedovalne modele in si ogledali njihove implementacije. Največ pozornosti smo posvetili modelu za napovedovanje panoge, zaradi predpostavke, da je od lastnosti, ki smo jih našteli, ta najbolj pomembna. Naš osnoven model je iz vseh posameznih strani spletnega mesta podjetja izluščil besedila in jih obravnaval kot en dokumet. S pomočjo tehnik NLP smo iz tega dokumenta pridobili značilke, ki smo jih uporabili v SVM klasifikatorju, kot lahko vidimo na Sliki 3.1. Poleg osnovnega modela smo implementirali še alternativne modele, ki so delovali na podoben način, le da so namesto celotnega besedila upoštevali le določene dele spletnih strani. Na koncu pa smo želeli preveriti tudi model, ki napoved podaja na podlagi napovedi ostalih omenjenih modelov. Za napovedovanje starosti in števila zaposlenih smo se odločili, da uporabimo drugačne značilke kot za napovedovanje panoge. Pokazali smo, katere alternativne značilke lahko izluščimo iz spletišč podjetij in uporabimo za napovedovanje. Pri teh dveh napovedovanjih je bilo zelo pomembno tudi ali bomo problem obravnavali kot klasifikacijo ali regresijo. Pri vseh napovednih modelih smo se posvetili tudi izbiranju primerne metrike uspešnosti, ki pravilno odraža naš cilj in upošteva naše vhodne podatke. Nato smo si v Poglavju 4 natančneje pogledali rezultate naših napovednih modelov in jih ovrednotili. Rezultate smo obravnavali tudi s stališča uporabnosti in predvidevali možnosti za še dodatne izboljšave.

V poslovnem svetu je potreba po podatkih o lastnostih podjetja velika. Cilj magistrskega dela je bil razviti napovedni model, ki bi lahko predstavljal dodaten vir teh informacij in ki bi se zanašal le na prosto dostopne podatke — v našem primeru spletno mesto podjetja. Cilj je bil deloma dosežen. Pokazali smo, da do neke mere tak model lahko uporabimo za dodaten vir informacij o lastnostih podjetja. Rešitev smo predstavili podjetju Datafy.it, pri katerem pravijo, da bi jim tak model zagotovo pomagal pri njihovem delovanju. Vendar pa so rezultati preslabi, da bi ta vir lahko smatrali kot izjemno zanesljiv. To seveda ni tako presenetljivo, ko pomislimo na vse možne spremenljivke, ki jih v sistem vnašamo, če so naš vhodni podatek spletna mesta. Kljub temu

pa so za specifične namene naši napovedni modeli uporabni. Kot smo videli v Poglavju 4, smo pri napovedovanju panoge za specifične panoge uspeli dobiti precej boljše rezultate, kot pa skupno povprečje 54.6%. Primer uporabe takih rezultatov bi bil recimo polavtomatsko iskanje podjetij. Model bi nam omogočal, da bi namesto pregledovanja vseh možnih podjetij v neki množici pregledali le manjšo množico. Za določitev te množice bi uporabili rezultate našega napovednega modela. A kaj takega pride v poštev le za tiste panoge, ki imajo v modelu zadosti visok priklic.
Podobno bi lahko izkoristili tudi model za napovedovanje števila zaposlenih. Predvsem bi se uporabnost izkazala za velikosti, pri katerih smo dosegli višji rezultat od povprečnega.
In podoben primer uporabe velja za napoved starosti podjetja. Tu bi bil verjetno bolj zanimiv klasifikacijski model, saj nam regresija deluje s povprečno absolutno napako 21.38 let. Glede na razpon starosti podjetij, ki jih obravnavamo, je to sicer nizka vrednost, vendar s praktičnega vidika v večini primerov povsem previsoka. Veliko bolj vsakdanji primer je določanje ali je podjetje staro ali mlado. Na žalost pa se je tudi klasifikacijski model v tem primeru odrezal bolj slabo. Dosegli smo točnost 41.7% in priklic mladih podjetij 72.4%. Predvidevamo, da so slabi rezultati tega modela predvsem posledica podatkov o starosti podjetja in kaj starost podjetja sploh pomeni oziroma predstavlja. Tega potencialnega problema smo se zavedali že ob načrtovanju modela v Poglavju 3.3.

V delu smo pokazali, da lahko zgolj na podlagi spletnega mesta relativno uspešno napovedujemo lastnosti, ki so s spletnim mestom povezane le posredno. Pri tem smo pokazali katere strukture in značilke so se izkazale za bolj ali manj uporabne. Orisali smo tudi kako lahko izvedemo tako napovedovanje in kateri koraki so ob tem ključni. Pokazali smo, kaj se pri takem napovedovanju lahko izkaže za problematično in s tem nakazali na prihodnje raziskave. Kljub omenjeni problematiki pa bi zagotovo lahko načine takega napovedovanja uporabili tudi v sorodnih problemih. Govorimo o problemih, pri katerih želimo le na podlagi spletnega mesta napovedati določeno lastnost.

Glede na količino spletnih mest in veliko potrebo po raznoraznih podatkih in avtomatiziranju pridobivanja le-teh, takih problemov oziroma priložnosti ni malo.

Dodatek A

Seznam podobnih panog

Panoga	Pripadajoče *podobne* panoge
Accounting	Financial Services
Airlines/Aviation	Aviation & Aerospace; Leisure, Travel & Tourism
Apparel & Fashion	Consumer Goods; Design; Luxury Goods & Jewelry; Retail
Architecture & Planning	Construction; Furniture
Automotive	Machinery; Mechanical or Industrial Engineering
Aviation & Aerospace	Airlines/Aviation; Leisure; Travel & Tourism; Mechanical or Industrial Engineering
Banking	Financial Services
Broadcast Media	Newspapers
Building Materials	Construction
Business Supplies and Equipment	Furniture; Printing; Retail
Chemicals	Cosmetics; Plastics

Civic & Social Organization	Government Administration; Nonprofit Organization Management
Civil Engineering	Architecture & Planning; Construction; Renewables & Environment
Computer Games	Information Technology and Services; Internet
Computer Software	Computer Games; Information Services; Information Technology and Services; Internet; Research
Construction	Architecture & Planning; Civil Engineering; Real Estate; Building Materials
Consumer Electronics	Business Supplies and Equipment; Electrical/Electronic Manufacturing; Entertainment; Information Technology and Services
Consumer Goods	Computer Games; Consumer Services; Food & Beverages; Wine and Spirits; Retail; Wholesale
Consumer Services	Hospitality; Leisure; Travel & Tourism; Retail; Wholesale
Cosmetics	Health, Wellness and Fitness; Hospital & Health Care
Design	Apparel & Fashion; Photography

E-Learning	Computer Games; Computer Software; Education Management; Information Technology and Services; Higher Education; Internet
Education Management	Management Consulting; Higher Education
Electrical/Electronic Manufacturing	Consumer Electronics; Industrial Automation
Entertainment	Apparel & Fashion; Broadcast Media; Computer Games; Events Services; Hospitality; Leisure, Travel & Tourism; Motion Pictures and Film; Music; Performing Arts
Environmental Services	Renewables & Environment
Events Services	Entertainment; Food & Beverages; Hospitality; Leisure, Travel & Tourism; Music; Restaurants
Financial Services	Accounting; Banking; Insurance; Legal Services
Food & Beverages	Events Services; Food Production; Restaurants; Wine and Spirits
Food Production	Consumer Goods; Food & Beverages; Wine and Spirits
Furniture	Architecture & Planning

Government Administration	Civic & Social Organization; Education Management; Environmental Services; Higher Education; Hospital & Health Care; Renewables & Environment
Graphic Design	Design; Printing
Health, Wellness and Fitness	Cosmetics; Hospital & Health Care; Hospitality; Leisure, Travel & Tourism
Higher Education	Education Management; Research
Hospital & Health Care	Health, Wellness and Fitness; Individual & Family Services; Medical Devices
Hospitality	Events Services; Leisure, Travel & Tourism
Human Resources	Government Administration; Professional Training & Coaching; Staffing and Recruiting
Industrial Automation	Mechanical or Industrial Engineering
Information Services	Computer Software; E-Learning; Information Technology and Services; Internet
Information Technology and Services	Computer Games; Computer Software; Information Services; Internet
Insurance	Banking; Financial Services
International Trade and Development	Government Administration

Internet	Computer Software; Information Services; Information Technology and Services
Investment Management	Financial Services; Insurance
Law Practice	Legal Services
Legal Services	Law Practice
Leisure, Travel & Tourism	Entertainment; Museums and Institutions
Luxury Goods & Jewelry	Apparel & Fashion
Machinery	Automotive; Mechanical or Industrial Engineering; Mining & Metals
Management Consulting	Professional Training & Coaching
Maritime	Leisure, Travel & Tourism
Mechanical or Industrial Engineering	Airlines/Aviation; Automotive; Industrial Automation; Plastics; Research; Machinery
Media Production	Broadcast Media; Motion Pictures and Film; Newspapers
Medical Devices	Hospital & Health Care
Mining & Metals	Building Materials; Machinery; Oil & Energy
Motion Pictures and Film	Entertainment; Media Production; Photography
Newspapers	Broadcast Media
Nonprofit Organization Management	Civic & Social Organization
Oil & Energy	Chemicals; Mining & Metals
Online Media	Entertainment; Media Production; Newspapers

Pharmaceuticals	Cosmetics; Hospital & Health Care
Printing	Business Supplies and Equipment
Professional Training & Coaching	Education Management; Human Resources; Management Consulting
Public Relations and Communications	Marketing and Advertising
Publishing	Media Production; Printing
Renewables & Environment	Environmental Services
Research	Higher Education
Restaurants	Food & Beverages; Hospitality
Retail	Apparel & Fashion
Security and Investigations	Information Technology and Services
Sports	Health, Wellness and Fitness; Leisure, Travel & Tourism
Staffing and Recruiting	Human Resources; Management Consulting
Telecommunications	Information Technology and Services
Transportation/Trucking/Railroad	Automotive; Logistics and Supply Chain

Tabela A.1: Seznam podobnih panog

Dodatek B

Podrobni rezultati napovedi panoge

Panoga	Utežen priklic	Št. vzorcev
Accounting	69.2%	1060
Airlines/Aviation	38.2%	526
Apparel & Fashion	63.3%	1482
Architecture & Planning	75.9%	2398
Automotive	72.7%	2343
Aviation & Aerospace	59.6%	540
Banking	72.2%	1042
Biotechnology	63.9%	898
Broadcast Media	32.9%	1011
Building Materials	31.8%	1083
Business Supplies and Equipment	14.1%	764
Chemicals	50.0%	760
Civic & Social Organization	34.7%	1309
Civil Engineering	46.9%	854
Computer Games	62.5%	591
Computer Software	47.3%	5652
Construction	64.1%	3568

Consumer Electronics	37.5%	726
Consumer Goods	25.4%	1371
Consumer Services	16.2%	735
Cosmetics	79.1%	593
Design	31.0%	3135
E-Learning	46.3%	724
Education Management	45.2%	2933
Electrical/Electronic Manufacturing	58.6%	2467
Entertainment	37.0%	1757
Environmental Services	48.8%	1479
Events Services	59.0%	3124
Facilities Services	18.2%	725
Financial Services	68.1%	4134
Food & Beverages	61.8%	1919
Food Production	51.7%	951
Furniture	56.8%	903
Government Administration	66.0%	1231
Graphic Design	18.9%	960
Health, Wellness and Fitness	65.2%	2300
Higher Education	80.9%	2717
Hospital & Health Care	69.3%	2572
Hospitality	78.0%	3645
Human Resources	43.5%	1816
Individual & Family Services	11.1%	560
Industrial Automation	30.2%	880
Information Services	31.9%	690
Information Technology and Services	69.4%	12388
Insurance	87.7%	1909
International Trade and Development	25.0%	816
Internet	26.1%	6760
Investment Management	34.4%	828

Law Practice	83.1%	1322
Legal Services	43.7%	1073
Leisure, Travel & Tourism	76.7%	4709
Logistics and Supply Chain	43.3%	1140
Luxury Goods & Jewelry	63.6%	548
Machinery	48.7%	1458
Management Consulting	26.0%	4861
Maritime	63.1%	648
Market Research	28.9%	669
Marketing and Advertising	79.0%	13181
Mechanical or Industrial Engineering	36.8%	2288
Media Production	18.8%	1881
Medical Devices	62.1%	1326
Mining & Metals	57.7%	573
Motion Pictures and Film	48.8%	635
Museums and Institutions	60.0%	529
Music	70.0%	996
Newspapers	30.6%	695
Nonprofit Organization Management	51.6%	4018
Oil & Energy	55.9%	1790
Online Media	6.1%	1055
Performing Arts	51.5%	583
Pharmaceuticals	46.3%	1022
Photography	60.0%	608
Plastics	50.0%	506
Printing	58.6%	1038
Professional Training & Coaching	44.2%	2206
Public Relations and Communications	63.5%	1491
Publishing	47.1%	2804
Real Estate	77.5%	3213
Renewables & Environment	64.6%	2074

Research	39.2%	1725
Restaurants	64.5%	751
Retail	38.7%	2977
Security and Investigations	56.4%	705
Sports	66.3%	1691
Staffing and Recruiting	77.8%	2855
Telecommunications	72.6%	2643
Transportation/Trucking/Railroad	53.2%	1067
Venture Capital & Private Equity	36.7%	561
Wholesale	12.7%	1173
Wine and Spirits	77.8%	549

Tabela B.1: Podrobni rezultati napovedi panoge

Literatura

[1] J. M. Pierre, "On the automated classification of web sites," *Linköping Electronic Articles in Computer and Information Science*, vol. 6, no. 1, 2006.

[2] D. Thorleuchter and D. V. den Poel, "Predicting e-commerce company success by mining the text of its publicly-accessible website," *Expert Systems with Applications*, vol. 39, no. 17, pp. 13026 – 13034, 2012.

[3] M. Kudelka, V. Snasel, Z. Horak, A. E. Hassanien, A. Abraham, and J. D. Velásquez, "A novel approach for comparing web sites by using microgenres," *Engineering Applications of Artificial Intelligence*, vol. 35, pp. 187 – 198, 2014.

[4] M. Sara, r. Amir masoud, and D. Mashallah Abbasi, "Webpage classification based on url features and features of sibling pages.," *International Journal of Computer Science and Information Security, Vol 8, Iss 2, Pp 168-173 (2010)*, no. 2, p. 168, 2010.

[5] M.-Y. Kan and H. O. N. Thi, "Fast webpage classification using url features," in *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, CIKM '05, (New York, NY, USA), pp. 325–326, ACM, 2005.

[6] R. Rajalakshmi and C. Aravindan, *Naive Bayes Approach for Website Classification*, pp. 323–326. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.

[7] X. Qi and B. D. Davison, "Web page classification: Features and algorithms," *ACM Comput. Surv.*, vol. 41, pp. 12:1–12:31, Feb. 2009.

[8] D. Shen, Z. Chen, Q. Yang, H.-J. Zeng, B. Zhang, Y. Lu, and W.-Y. Ma, "Web-page classification through summarization," in *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, (New York, NY, USA), pp. 242–249, ACM, 2004.

[9] R.-C. Chen and C.-H. Hsieh, "Web page classification based on a support vector machine using a weighted vote schema," *Expert Systems with Applications*, vol. 31, no. 2, pp. 427 – 435, 2006.

[10] O.-W. Kwon and J.-H. Lee, "Text categorization based on k-nearest neighbor approach for web site classification," *Information Processing & Management*, vol. 39, no. 1, pp. 25 – 44, 2003.

[11] M. Ester, H.-P. Kriegel, and M. Schubert, "Web site mining: A new way to spot competitors, customers and suppliers in the world wide web," in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, (New York, NY, USA), pp. 249–258, ACM, 2002.

[12] J.-C. Lamirel and D. Reymond, "Automatic websites classification and retrieval using websites communication signatures," *COLLNET Journal of Scientometrics and Information Management*, vol. 8, no. 2, pp. 293–310, 2014.

[13] A. Bergholz, J. H. Chang, G. Paass, F. Reichartz, and S. Strobel, "Improved phishing detection using model-based features.," in *CEAS*, 2008.

[14] H. Misra, F. Yvon, O. Cappé, and J. Jose, "Text segmentation: A topic modeling perspective," *Information Processing & Management*, vol. 47, no. 4, pp. 528 – 544, 2011.

[15] I. Vulić, W. D. Smet, J. Tang, and M.-F. Moens, "Probabilistic topic modeling in multilingual settings: An overview of its methodology and applications," *Information Processing & Management*, vol. 51, no. 1, pp. 111 – 147, 2015.

[16] T. Jorge Victor Carrera, S. Grigori, M.-J. Sabino, I. Marco Moreno, and M. Rodrigo Cadena, "Latent dirichlet allocation complement in the vector space model for multi-label text classification.," *International Journal of Combinatorial Optimization Problems and Informatics, Vol 6, Iss 1, Pp 7-19 (2015)*, no. 1, p. 7, 2015.

[17] C. Bauer and A. Scharl, "Quantitive evaluation of web site content and structure," *Internet Research*, vol. 10, no. 1, pp. 31–44, 2000.

[18] U. S. Census, "North american industry classification system," 2016. Dostopno na `http://www.census.gov/eos/www/naics/` (dostopano 20.12.2016).

[19] Wikipedia, "Cross industry standard process for data mining," 2016. Dostopno na `https://en.wikipedia.org/wiki/Cross_Industry_Standard_Process_for_Data_Mining` (dostopano 1.1.2017).

[20] F. Pérez and B. E. Granger, "IPython: a system for interactive scientific computing," *Computing in Science and Engineering*, vol. 9, pp. 21–29, May 2007.

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[22] LinkedIn, "Industry codes," 2015. Dostopno na `https://developer.linkedin.com/docs/reference/industry-codes` (dostopano 20.10.2015).

[23] U. N. S. Division, "International standard industrial classification of all economic activities," 2017. Dostopno na http://unstats.un.org/UNSD/cr/registry/regcst.asp?Cl=2 (dostopano 20.10.2015).